

SUPER

L. 7.500
Settembre 1986-Anno 3-n°9

COMMODORE 64 e 128

SUPERCOMMODORE 64
La rivista per gli utenti di
COMMODORE 64 e 128

AUTODISK

SCREEN GENIE

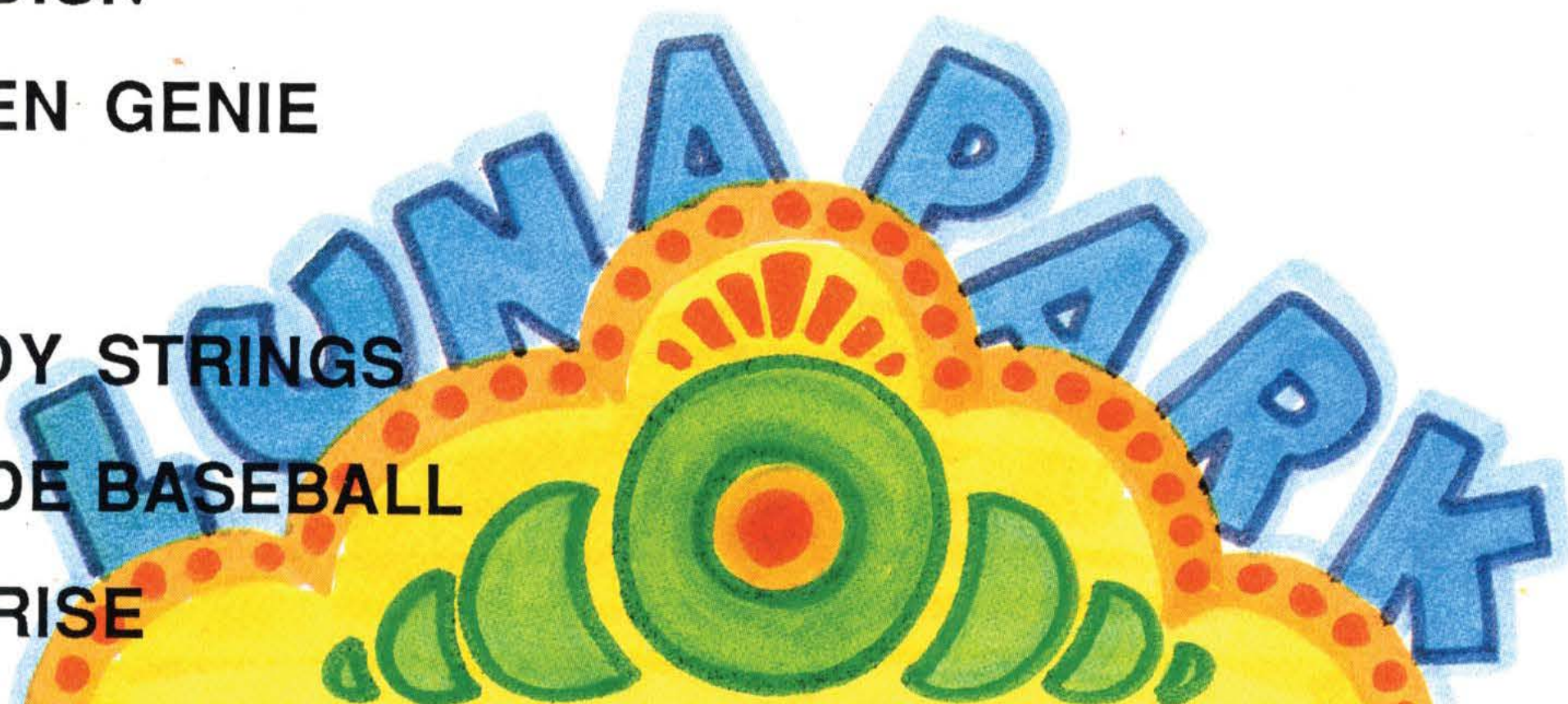
HELP!

SPEEDY STRINGS

ARCADE BASEBALL

HIGH RISE

SOLARPIX



TIRO A SEGNO



solo
L.9.900

COLLEZIONE

SUPER

COMMODORE
C-64
C-16

5

e VIC-20

oltre

20
programmi

GIOCHI
UTILITYES
TOOL

3

riviste



GRUPPO EDITORIALE
JACKSON

DIVISIONE PERIODICI

I.V.A. assolta dall'Editore
(art. 74, 1° comma - lettera c) D.P.R. 633/72

SUPER

COMMODORE 64 e 128

4 **READ & WRITE**

LA POSTA DEI LETTORI

9 **OPEN**

COME RIPOSIZIONARE LA
MEMORIA BASIC DEL C64
di M.Cristuib Grizzi

12 **LA CULTURA IS INFORMATICA**
...then

UNO SGUARDO ALLA
GRAFICA DEL C64
Quarta Parte
di J. Heilborn
trad. e adatt.
di E. Stefani

17 **WORK**

17 AUTODISK
di M.Cristuib Grizzi

20 SCREEN GENIE
di J.A. Ledger
trad. e adatt.
di L. Priotto

32 HELP!
di J. Siddiqui
trad. e adatt.
di E. Stefani

36 SPEED SCRIPT 80
PER C128
di T. Heimark
trad. e adatt.
di S. Colombo

40 SPEEDY STRINGS
di T. Friedman
trad. e adatt.
di S. Colombo

43 **ARC**

43 ARCADE BASEBALL
di K. Mykytyn e M. Tuttle
trad. e adatt.
di E. Stefani

48 HIGH RISE
di C. Mc Guyer
trad. e adatt.
di L. Priotto

57 SOLARPIX
di S. Edgeworth
trad. e adatt.
di S. Colombo

61 **P R I N T**

RECENSIONI LIBRI

65 **input-output**

PICCOLI ANNUNCI

GRUPPO EDITORIALE
JACKSON s.p.a.
San Francisco - Londra -
Madrid - Milano

DIREZIONE, REDAZIONE,
AMMINISTRAZIONE
Via Rosellini, 12
20124 MILANO
Tel. (02) 68.80.951/2/3/4/5
Telex 333436 GEJIT I

SEDE LEGALE
Via G. Pozzone, 5 - 20121 Milano

CONSOCIATE ESTERE USA
GEJ Publishing Group, Inc.
a subsidiary of
Gruppo Editoriale Jackson
1307 S. Mary Avenue - 94087
Sunnyvale CA
Tel. (408) 7730138 - Telex 49959972

GRAN BRETAGNA
GEJ Publishing Ltd.
187 Oxford Street London WURIAJ
Tel. (01) 4392931 - Telex (051)
21248

DIREZIONE EDITORIALE
Daniele Comboni

DIREZIONE AMMINISTRATIVA
Giuliano Di Chiano

DIREZIONE DIVISIONE LIBRI
E GRANDI OPERE
Roberto Pancaldi

DIRETTORE RESPONSABILE
Giampietro Zanga

REDAZIONE
Mauro Cristuib Grizzi

HANNO COLLABORATO A
QUESTO NUMERO

S. Colombo
M. Cristuib Grizzi
L. Priotto
E. Stefani

GRAFICA E IMPAGINAZIONE
Ivana Rossi

STAMPA
Litografia del Sole
20080 Albairate (MI)

PUBBLICITA'
Concessionaria per l'Italia e l'Estero
J. Advertising s.r.l.
Viale Restelli, 5 - 20124 Milano
Tel. (02) 68.82.895-68.82.48-
68.80.606
TELEX 316213 REINA I

Concessionaria per USA e Canada
International Media Marketing
16704 Margardt Avenue
P.O. Box 1217 - Cerritos CA 90701
Tel. (213) 926 - 9552

DISTRIBUZIONE
SODIP
Via Zuretti, 25 - 20125 Milano
Spedizione in abbonamento postale
Gruppo III/70
Pubblicità inferiore al 70%
Prezzo della rivista L. 7.500
Numero arretrato L. 15.000
Abbonamento L. 82.500 (11 numeri
con cassetta); L. 38.500 (11 numeri
senza cassetta)
per l'Estero L. 110.000 (11 numeri
con cassetta); L. 50.000 (11 numeri
senza cassetta)

I versamenti vanno indirizzati a:
Gruppo Editoriale Jackson
Via Rosellini, 12 - 20124 Milano
mediante emissione di assegno
bancario, vaglia o utilizzando il
C.C.P. n° 11666203
Per i cambi di indirizzo indicare, oltre
al nuovo, anche l'indirizzo
precedente e allegare L. 500, anche
in francobolli.

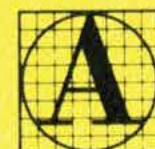
AUTORIZZAZIONE ALLA
PUBBLICAZIONE
Tribunale di Milano
n° 155 del 5/4/86

GLI ARTICOLI TRADOTTI SONO
TRATTI DALLE RIVISTE
COMPUTER O DA
COMPUTER GAZETTE
COMPUTER PUBLICATION, INC.
P.O. Box 5406, Greensboro, NC
27403 - USA

Il Gruppo Editoriale Jackson è
iscritto nel Registro nazionale
della Stampa al n° 117 vol. 2 -
foglio 129 in data 17-8-1982

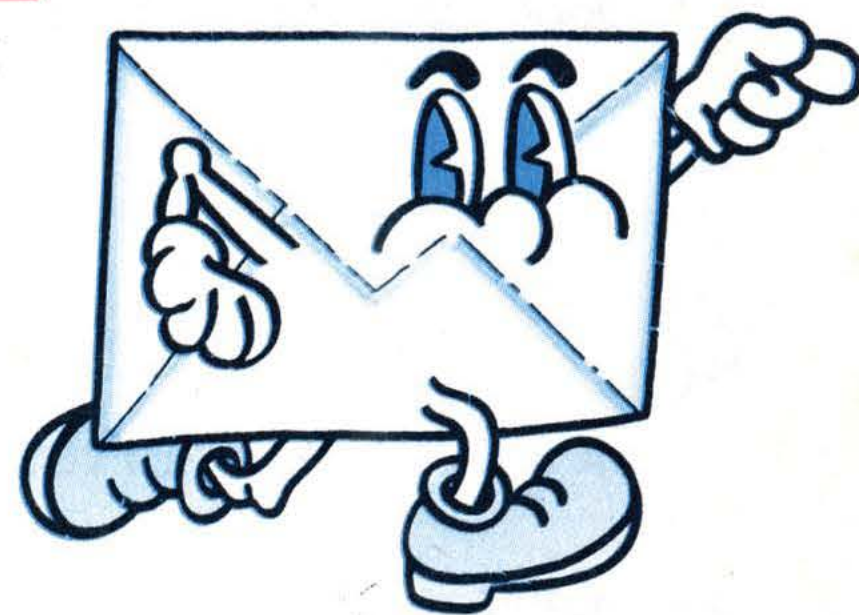


Rivista associata
all'Unione
Stampa
Periodica
Italiana



ASSOFT Associata

READ & WRITE



Leggere la directory

Ho visto in alcuni programmi pubblicati delle routine per la lettura della directory del disco e la visualizzazione dei file in esso contenuti. Non mi è tuttavia molto chiaro il funzionamento di dette routine, soprattutto delle molte istruzioni CHR\$ che sono sempre presenti. Potreste essermi d'aiuto?

D. Rigamonti

R. Ecco un esempio, del tutto generale, di una routine di lettura della directory del dischetto:

```
10 OPEN2,8,0,"$"
20 GET#2,A$:GET#2,A$
30 GET#2,L$:GET#2,L$:IFST=64
  THENCLOSE2:END
40 GET#2,LB$:GET#2,HB$:LN=ASC
  (LB$+CHR$(0))+256*ASC
  (HB$+CHR$(0))
50 PRINT LN;
60 GET#2,A$:IFA$=""THENPRINT
  CHR$(13);:GOTO30
70 PRINTA$;:GOTO60
```

Il file della directory, che ha nome "\$", può essere letto proprio come qualsiasi altro file su disco: dal momento che è un file in formato programma, occorre utilizzare l'indirizzo secondario 0 (buffer 0) nell'istruzione OPEN di linea 10.

Come sempre, i primi due byte di un file programma rappresentano il suo indirizzo iniziale in formato byte basso / byte alto; poiché non interessano ai nostri scopi, essi sono letti, ma ignorati dalla linea 20.

Ogni linea di un programma BASIC è composta in RAM e su disco da quattro parti: il link di linea, il numero di linea, il contenuto della linea e uno zero finale.

La linea 30 legge il link e controlla il contenuto della variabile riservata ST: se

questa assume il valore 64, si è giunti alla fine del file, il quale viene chiuso con termine della routine.

Pertanto, se intendete utilizzare questa routine nei vostri programmi, rinumeratela opportunamente e sostituite con un RETURN l'istruzione END in linea 30.

La linea 40 legge i byte basso e alto del numero di linea e ne calcola il valore decimale, che viene stampato sullo schermo dalla linea 50.

Naturalmente, la directory di un disco non possiede numeri di linea veri e propri, ma, trattandosi di un file programma, il numero di blocchi occupati su disco da ciascun file viene trattato esattamente come il numero di linea di un'istruzione BASIC (vi sarà sicuramente capitato di dare inavvertitamente un RUN a una directory, ottenendo in risposta il messaggio "? SYNTAX ERROR IN 0"...). Il resto della linea è letto, carattere per carattere, fino a che viene raggiunto lo zero di fine linea, in corrispondenza del quale la routine stampa sullo schermo un CHR\$(13), cioè un carattere di RETURN.

Dopo l'andata a capo viene letta la linea seguente, e così via fino a che viene raggiunta la fine del file.

Se desiderate inserire i nomi di file all'interno di un vettore, è sufficiente concatenare in un'unica stringa i caratteri letti, dopo la GET#2,A\$ di linea 60.

Relfiling e i caratteri speciali nei nomi di file

Ho notato che i nomi dei file salvati su disco dal programma "Relfiling System 2.0" appaiono sulla directory con uno strano carattere in "reverse", mentre se caricati dal programma stesso lo strano

simbolo scompare nella visualizzazione del nome. E' normale tutto ciò?

D. Marini

R. E' perfettamente normale: il programma aggiunge infatti in testa al nome del file un CHR\$(18), ovvero un carattere di REVERSE ON. Ciò consente al programma di riconoscere, tra i vari file eventualmente presenti sul dischetto, quali sono gli archivi e gli indici da utilizzare, e quali invece normali file programma o dati generati da altri programmi.

Oltre a ciò, la visualizzazione da programma del nome del file appare già automaticamente in campo inverso.

A proposito di Relfiling System 2.0, ci è stato segnalato che su alcune (per fortuna poche) cassette il programma non viene trovato dal registratore, anche se è chiaramente presente dato il lungo scorrere del nastro. Pur non essendo ancora riusciti a spiegarci il perché dell'inconveniente, abbiamo verificato che Relfiling risulta perfettamente caricabile, se si carica il programma che lo precede (Preview 80) con il doppio 1, si dà un NEW a fine caricamento e si digita poi un semplice LOAD. A questo punto Relfiling verrà caricato da cassetta e sarà pronto per essere trasferito su disco.

Si noti infine che caricando Relfiling da nastro ed attivandolo prima di averlo trasferito su disco, la visualizzazione della data corrente risulta distorta: ciò a causa dell'utilizzo del buffer del registratore per la memorizzazione dei parametri relativi alla data; l'inconveniente tuttavia scompare non appena si salvi su disco il programma stesso.

Dissolvenze sulle pagine grafiche

M. Miranetto

caricamento della successiva. Se la mappa di bit non viene più disattivata, il caricamento della seconda pagina grafica potrà essere "visto" a video, con l'immagine che pian piano si va a sovrapporre alla precedente. Poichè la locazione di memoria più bassa di una pagina grafica è la superiore sinistra, e la più alta è l'inferiore destra, l'immagine inizierà a crearsi dalla riga di pixel più alta per terminare in quella più bassa.

Speedscript e RS-232

stampa nulla, per cui vi sarei grato di una spiegazione dell'inconveniente.

R. Bissolati

R. Il programma SpeedScript non supporta l'uscita su RS-232, ma solo sul bus seriale: regolando semplicemente il numero di periferica sul 2 e l'indirizzo secondario su zero non si realizza un'apertura di canale RS-232, in quanto mancano così i parametri relativi ai registri di comando e di controllo (baud rate, parità, lunghezza parola, bit di stop, ecc.)

Se si possiede una stampante RS-232, occorre quindi ricorrere a un altro programma di word processing scritto per funzionare anche con questa interfaccia, come ad esempio il diffusissimo Easy Script. I file scritti con SpeedScript possono essere agevolmente convertiti nel formato richiesto da Easy Script (Commodore ASCII) attraverso il programma di conversione pubblicato insieme alla versione 3.0 di SpeedScript.

SUPERCOMMODORE è bello, però... (ovvero suggerimenti, idee, critiche, richieste e tutto ciò che vi passa per la testa).

[illegible]

Nome
Cognome
Via C.A.P.
Città Tel.

SUPERCOMMODORE

La rubrica INPUT/OUTPUT è gratuita ed aperta a tutti i lettori. Chi desidera comprare, vendere o cambiare hardware o software può inviare il tagliando a Gruppo Editoriale Jackson - Via Rosellini, 12 - 20124 MILANO

INPUT/OUTPUT

☐ COMPRO ☐ VENDO ☐ CAMBIO
☐ VIC 20 ☐ C 16 ☐ C64 ☐ PERIF. ☐ SOFTWARE

.....

.....

.....

.....

.....

Nome Cognome
Via C.A.P.
Città Tel.

OK-LIST

Sappiamo tutti quanto sia difficile digitare correttamente un programma (ovviamente, se non si dispone della cassetta allegata alla rivista!). Errori di digitazione anche banali possono provocare il malfunzionamento del programma stesso o anche il blocco completo del computer, obbligando a riprendere da capo tutto il lavoro.

OK-LIST è un particolare programma in grado di intercettare gli errori commessi durante la digitazione di uno dei listati da noi pubblicati, e di segnalarveli.

Digitate il listato pubblicato in questa pagina (attenzione a non commettere errori: OK-LIST non può controllare se stesso!) e quindi salvatelo su nastro o disco.

Quando dovrete accingervi a digitare uno dei nostri programmi in BASIC, caricate per prima cosa OK-LIST e dategli il RUN: dopo una breve attesa, necessaria a inserire in memoria una routine in linguaggio macchina, comparirà sullo schermo il messaggio "OK-LIST ATTIVATO".

Potete a questo punto iniziare a ricopiare il programma BASIC prescelto, evitando di digitare la "rem" seguita da un numero che compare al termine di ogni linea. Come vedrete, alla pressione di RETURN apparirà un numero nell'angolo superiore sinistro dello schermo; confrontatelo con quello riportato dopo la "rem" sul listato: se è diverso, avete commesso un errore nel digitare quella linea di programma.

OK-LIST non verifica gli spazi all'interno di una linea di istruzione: se tutti i checksum (così vengono chiamati i numerini di controllo visualizzati da OK-LIST) sono corretti, e il programma genera invece delle videate 'spezzettate', probabilmente avete dimenticato o aggiunto qualche spazio di troppo all'interno delle virgolette di una istruzione PRINT.

Inoltre, OK-LIST non è in grado di controllare la sequenza con la quale vengono digitati i caratteri all'interno di un'istruzione; ciò è particolarmente pericoloso nel caso delle istruzioni DATA contenenti molti numeri. Ad esempio, digitando 220 dove andrebbe digitato 202, non si verifica alcuna variazione nel checksum, pur avendo commesso un errore; consi-

gliamo quindi di porre grande attenzione nel digitare tale tipo di istruzioni.

Un'ultima avvertenza sull'utilizzo di OK-LIST riguarda le abbreviazioni delle istruzioni BASIC: esse non vengono riconosciute da OK-LIST e generano pertanto un checksum differente. Se si utilizza OK-LIST, le istruzioni BASIC dovranno quindi essere digitate per intero.

Data la sua posizione in memoria (il buffer del registratore), OK-LIST interferisce con ogni operazione con l'unità a cassette. Per questo motivo prima di effettuare un SAVE su nastro OK-LIST dovrà essere preventivamente disabilitato attraverso la pressione di RUN/STOP e RESTORE.

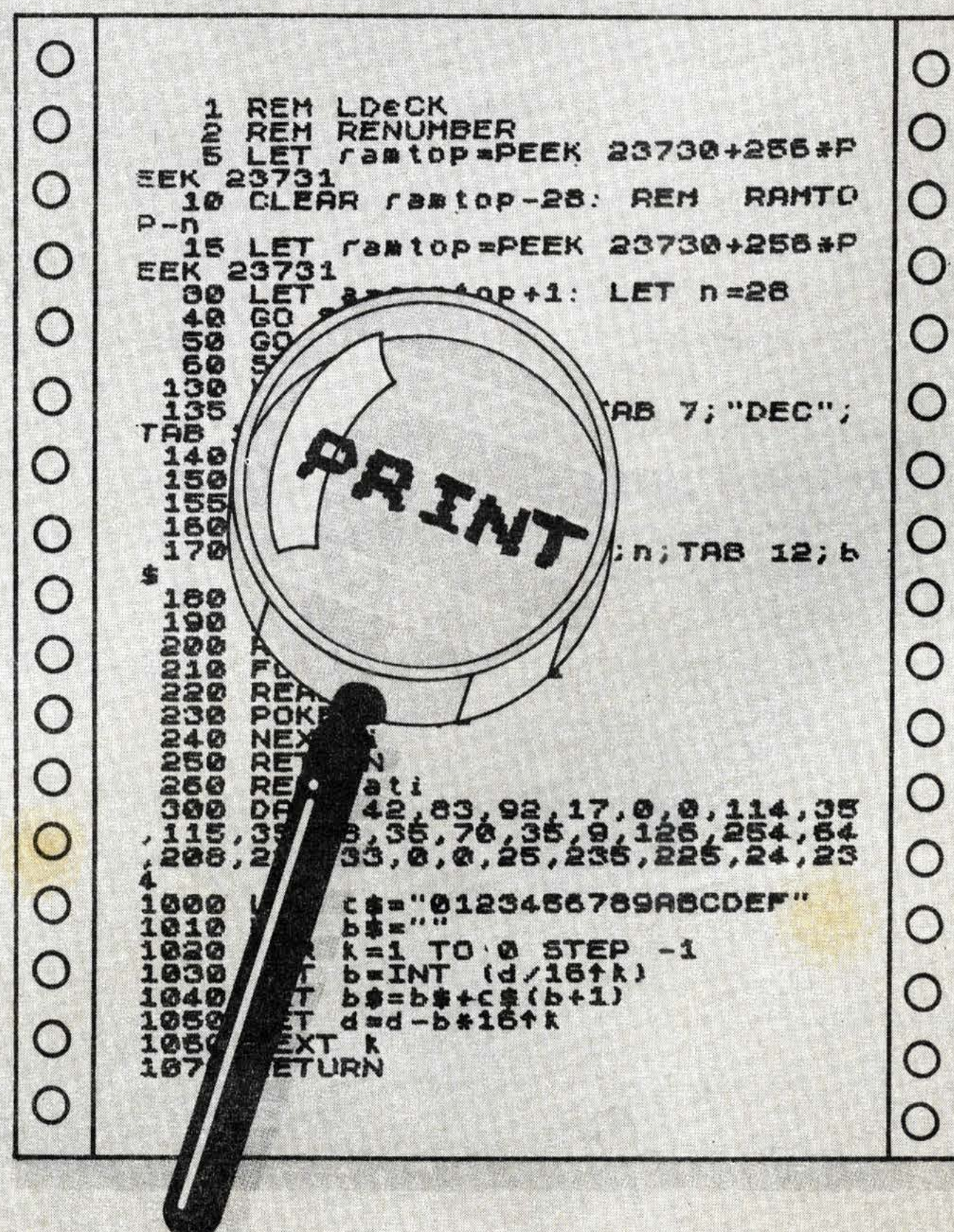
Non è possibile effettuare operazioni di LOAD da cassetta senza cancellare OK-LIST dalla memoria.

Queste limitazioni non sono invece presenti, se si utilizza il disk drive come memoria di massa.

Nel caso si prema accidentalmente RUN/STOP e RESTORE durante l'utilizzo di OK-LIST, esso potrà essere riattivato digitando SYS886.

Buon lavoro!

```
10 PRINT "{CLR}ATTENDERE PREGO..."
11 FOR I=886 TO 1018: READ A: CK=CK+A: POKE I, A: NEXT I
12 IF CK <> 17539 THEN PRINT "{GIU'}ERRORE NELLE ISTRUZIONI DATA": END
13 SYS886: PRINT "{CLR}{ 2 GIU'}OK-LIST ATTIVATO.": NEW
14 DATA 173,036,003,201,150,208
15 DATA 001,096,141,151,003,173
16 DATA 037,003,141,152,003,169
17 DATA 150,141,036,003,169,003
18 DATA 141,037,003,169,000,133
19 DATA 254,096,032,087,241,133
20 DATA 251,134,252,132,253,008
21 DATA 201,013,240,017,201,032
22 DATA 240,005,024,101,254,133
23 DATA 254,165,251,166,252,164
24 DATA 253,040,096,169,013,032
25 DATA 210,255,165,214,141,251
26 DATA 003,206,251,003,169,000
27 DATA 133,216,169,019,032,210
28 DATA 255,169,018,032,210,255
29 DATA 169,058,032,210,255,166
30 DATA 254,169,000,133,254,172
31 DATA 151,003,192,087,208,006
32 DATA 032,205,189,076,235,003
33 DATA 032,205,221,169,032,032
34 DATA 210,255,032,210,255,173
35 DATA 251,003,133,214,076,173
36 DATA 003
```



Piccola guida per l'input dei programmi

I programmi in BASIC listati su **SUPER-COMMODORE** contengono una particolare codifica dei caratteri di controllo (cursore, colore, reverse, eccetera) che permette di ottenere una maggiore leggibilità dei programmi.

Generalmente, questi listati contengono alcuni caratteri racchiusi tra parentesi graffe { }; queste ultime, che non esistono sulla tastiera del computer, non devono essere digitate, ma hanno unicamente lo scopo di indicare che i caratteri da esse racchiusi sono dei caratteri di controllo.

Ad esempio, {GIU'} indica che occorre premere una volta il tasto di cursore verso il basso, {3 GIU'} indica che il tasto di cursore verso il basso dovrà essere premuto tre volte.

Se tra le parentesi graffe è racchiuso un singolo carattere, quest'ultimo deve essere premuto insieme al tasto CTRL (ad esempio, incontrando {A} si dovrà premere il tasto CTRL

insieme al tasto A).

Invece i caratteri racchiusi tra parentesi quadre e simboli di maggiore e minore [< >] devono essere premuti unitamente al tasto Commodore (nell'angolo inferiore sinistro della tastiera). Ad esempio, [<A>] indica che deve essere premuto il tasto Commodore insieme al tasto A.

Infine, alcuni caratteri racchiusi dalle parentesi graffe preceduti dalle lettere SH: ciò indica che il carattere seguente deve essere digitato tenendo premuto il tasto SHIFT.

La tabella seguente riporta, per ogni carattere di controllo, la codifica utilizzata nei nostri listati.

I listati in linguaggio macchina (totalmente numerici) non possono invece essere copiati direttamente con il computer, ma richiedono l'utilizzo dello speciale programma MLX, riportato, insieme alle istruzioni per il suo utilizzo, in altra parte della rivista.

Quando leggete:	Premete:	Vedrete:	Quando leggete:	Premete:	Vedrete:	Quando leggete:	Premete:	Vedrete:
{CLR}	SHIFT CLR/HOME		{CYN}	CTRL 4		[<7>]	G 7	
{HOME}	CLR/HOME		{PUR}	CTRL 5		[<8>]	G 8	
{SU}	SHIFT ↑ CRSR ↓		{GRN}	CTRL 6		{F1}	f1	
{GIU'}	↓ CRSR ↓		{BLU}	CTRL 7		{F2}	f2	
{SIN}	SHIFT ⇐ CRSR ⇒		{YEL}	CTRL 8		{F3}	f3	
{DES}	⇐ CRSR ⇒		[<1>]	G 1		{F4}	f4	
{RVS}	CTRL 9		[<2>]	G 2		{F5}	f5	
{OFF}	CTRL 0		[<3>]	G 3		{F6}	f6	
{BLK}	CTRL 1		[<4>]	G 4		{F7}	f7	
{WHT}	CTRL 2		[<5>]	G 5		{F8}	f8	
{RED}	CTRL 3		[<6>]	G 6				

È in edicola PC World Magazine di settembre: con un favoloso dischetto DataLife e demo Lotus



**Un numero
da non perdere**



**GRUPPO EDITORIALE
JACKSON**
DIVISIONE PERIODICI



di M. Cristuib Grizzi

Come riposizionare la memoria BASIC del C64

Nella programmazione o anche solo nell'utilizzo di programmi già pronti, ci si trova spesso a dover spostare l'inizio o la fine della memoria BASIC per lasciare spazio a dati o programmi. In questo articolo esaminiamo tali tecniche, facendo riferimento a programmi già pubblicati.

Abbiamo più volte affrontato, nella rubrica *Read&Write*, l'argomento del trasferimento su dischetto dei nostri programmi in linguaggio macchina attraverso MLX. Malgrado ciò, una tecnica che appare essere piuttosto ostica per i programmatori è quella del riposizionamento della memoria BASIC, spesso richiesta dall'utilizzo di MLX, che si concretizza in una serie di istruzioni POKE da effettuare prima del caricamento del programma.

Analogamente, ci sono pervenute molte richieste di ulteriori spiegazioni sul calcolo di queste POKE di riposizionamento, necessarie anche per l'utilizzo del programma "Rillocatore" pubblicato sul numero 2 di quest'anno.

Vediamo dunque a che cosa servono queste famigerate POKE, e come calcolarle per ogni esigenza.

MLX è un programma che, oltre ad essere indispensabile per digitare i programmi in linguaggio macchina, è molto utile anche per trasferirli da nastro a disco e viceversa.

In entrambi i casi la memoria del C64 deve quindi ospitare contemporaneamente sia il programma MLX che il program-

Come riposizionare la memoria basic del C64



ma in linguaggio macchina: è pertanto fondamentale che non si verifichino in alcun modo sovrapposizioni, ma ciascun programma occupi zone di memoria distinte e separate.

I programmi in linguaggio macchina per il C64 possono essere suddivisi in quattro categorie, in funzione dell'area di memoria da essi occupata:

1. Programmi esterni all'area BASIC, ma non sottostanti la ROM (tipicamente quelli che partono dalla locazione 49152).
2. Programmi esterni all'area BASIC, ma sottostanti la ROM (anche se piuttosto rari, in genere partono da 40960 e stanno sotto l'interprete BASIC).
3. Programmi nell'area BASIC bassa (partono dalla locazione 2049 e in genere posseggono una linea BASIC che esegue la SYS di attivazione, si possono caricare e salvare senza accorgimenti particolari e come normali programmi BASIC).
4. Programmi nell'area BASIC alta (in ge-

nere il loro indirizzo iniziale è maggiore o uguale a 32768).

Ciascuna di queste categorie richiede un "trattamento" diverso per MLX, al fine di posizionarlo in una zona della memoria BASIC che non interferisca con il linguaggio macchina. In particolare quelli del gruppo 1, essendo già in un'area di memoria completamente separata da quella BASIC in cui risiede MLX, non richiedono alcuna POKE di riposizionamento per quest'ultimo.

I programmi del secondo gruppo sono alquanto rari (tra quelli da noi pubblicati c'è finora soltanto Ultrawedge) e, pur non interferendo con la memoria BASIC, richiedono una versione speciale di MLX, in grado di leggere il contenuto della RAM "nascosta" sotto la ROM del C64. Anche qui tuttavia non servono POKE di riposizionamento.

I programmi del terzo gruppo richiedono MLX soltanto per la loro digitazione, mentre per il trasferimento su disco o su un'altra cassetta è sufficiente una normale istruzione SAVE, in quanto si comportano come normali programmi BASIC.

Se si deve inserire tramite MLX uno di questi programmi, si noterà come sia il programma da digitare (e che quindi viene "creato", numero dopo numero, in memoria) sia MLX partano esattamente dalla stessa locazione 2049, corrispondente all'inizio dei programmi BASIC.

Poichè il programma in linguaggio macchina non si può spostare dalle sue locazioni "naturali", sarà necessario spostare MLX in una zona di memoria più alta dell'ultima locazione occupata dal linguaggio macchina.

I puntatori nelle locazioni 43 e 44 stabiliscono dove inizia la memoria BASIC.

All'accensione del computer la locazione 43 contiene uno zero, mentre la locazione 44 contiene il valore 8: cosa significa? Per capirlo è sufficiente utilizzare la solita formuletta per convertire il valore dei puntatori nell'indirizzo "puntato":

$$\text{indirizzo} = \text{byte basso} + \text{byte alto} * 256$$

avremo quindi, sostituendo i valori effettivi:

$$\text{indirizzo} = \text{PEEK}(43) + \text{PEEK}(44) * 256 = 0 + 8 * 256 = 2048$$

Come mai 2048, quando invece diciamo sempre che i programmi in BASIC partono dalla locazione 2049? La spiegazione è che il sistema operativo del C64 richiede sempre uno zero come primo byte, all'inizio di un programma BASIC; la locazione 2048 conterrà quindi uno

zero e la 2049 il primo byte del programma BASIC.

Ritornando ora a MLX e al nostro programma in linguaggio macchina, supponiamo che quest'ultimo termini alla locazione 6234: potremo fare in modo che MLX venga caricato a partire dalla locazione 8192, ben al di sopra della zona occupata dal primo. Poichè 8192 diviso 256 fa esattamente 32, e la locazione 43 già contiene uno zero all'accensione del computer, dovremo digitare la seguente linea subito dopo l'accensione del C64:

`POKE44,32:POKE32*256,0:NEW`

La prima POKE eleva il contenuto del puntatore 44 dal valore originario 8 a quello attuale di 32: applicando la formuletta precedente avremo che l'indirizzo ora "puntato" è:

$$\text{indirizzo} = \text{PEEK}(43) + \text{PEEK}(44) * 256 = 0 + 32 * 256 = 8192$$

Quindi dopo la prima POKE il BASIC inizia dalla locazione 8192. La seconda POKE provvede a inserirvi lo zero necessario al sistema operativo, mentre la NEW provvede a resettare di conseguenza anche gli altri puntatori.

Caricando ora in memoria MLX, esso occuperà le locazioni a partire dalla 8193, lasciando quindi completamente libero tutto lo spazio necessario al programma in linguaggio macchina.

I programmi del quarto gruppo, infine, non interferiscono direttamente con MLX, ma con il contenuto delle sue variabili. Se un programma BASIC risiede nella parte bassa dell'area BASIC, le sue variabili risiedono nella parte alta: ciò significa che se non si prendono le opportune contromisure, alcuni byte del programma in linguaggio macchina verranno sovrascritti dalle variabili utilizzate da MLX.

Se prima abbiamo spostato verso l'alto l'inizio della memoria BASIC, questa volta dovremo sportarne verso il basso il punto finale.

La memoria BASIC normalmente termina con la locazione 40959, e questo valore è controllato dai puntatori nelle locazioni 643 e 644. La modifica di questi puntatori, per essere accettata dal sistema operativo, richiede inoltre una SYS a una routine di reset di sistema.

Supponendo che il programma in linguaggio macchina parta dalla locazione 32768, potremo ad esempio spostare verso il basso la cima della memoria BASIC fino alla locazione 32256 (come avrete notato, è sempre conveniente utilizzare multipli di 256!). Poichè 32256

diviso 256 fa 126, daremo le seguenti istruzioni subito dopo aver acceso il C64:

`POKE643,0:POKE644,126:SYS58260`

La SYS58260 eseguirà il parziale reset, rivisualizzando la schermata iniziale con un numero di byte liberi congruente con la nuova cima della memoria BASIC. Potremo ora caricare normalmente MLX, con la certezza che non interferirà in alcun modo con le locazioni superiori alla 32256, dove si trova il programma in linguaggio macchina.

Come si vede, la gestione della memoria del C64 (e di tutti i computer Commodore) è molto flessibile, e permette interventi diretti da parte dell'operatore per riconfigurare secondo necessità tutta la mappa di memoria.

Rillocatore e riposizionamento della memoria BASIC

Le due linee di POKE ora viste vengono ampiamente utilizzate anche nell'uso del programma "Rillocatore" pubblicato sul numero 2 di quest'anno, anche se la situazione è più complessa: mentre utilizzando MLX sono soltanto due i programmi che devono risiedere contemporaneamente in memoria, con il Rillocatore sono ben tre.

La sovraffollata memoria del C64 deve infatti ospitare lo stesso Rillocatore, il programma da rillocare e il programma rillocato.

Poichè il programma rillocato non esiste a priori, ma viene creato per azione del Rillocatore, bisognerà prevederne lo spazio occupato e modificare i puntatori in modo che esso rimanga libero.

L'occupazione di memoria richiesta dal programma Rillocatore e da un programma da rillocare che inizia alla locazione X e termina alla locazione Y è la seguente:

1. dalla locazione 2049 fino alla locazione F, data dalla formula:

$$F = 500 + 2049 + (Y - X)$$

si dovrà liberare la memoria per ospitare il programma rillocato (500 è un coefficiente di sicurezza che tiene conto dell'aumentata lunghezza del programma rillocato dovuta alla routine di trasferimento, alla SYS iniziale, eccetera).

2. dalla locazione F+1 fino alla locazione X (se X < 40960) ci sarà lo spazio libero per il Rillocatore e le sue variabili. Se X è maggiore di 40960, lo spazio per il Rillocatore andrà da F+1 fino a 40960.

3. dalla locazione X alla locazione Y ci sarà il programma in linguaggio macchina da rilocare.

Come si vede, e salvo casi del tutto speciali e infrequenti (che non esaminiamo per maggior chiarezza), il Rilocatore si deve sempre trovare in una zona centrale di memoria, tra il programma rilocato e quello da rilocare.

I valori da inserire nei puntatori 43 e 44 - che chiamiamo rispettivamente A e B - saranno dati da:

$$B = \text{INT}((F+1)/256)$$

$$A = (F+1) - (B*256)$$

e l'istruzione che riserva memoria per il programma rilocato sarà dunque:

POKE43,A:POKE44,B:POKE
(A+B*256),0:NEW

Questa istruzione (che va data ovviamente prima di caricare in memoria il Rilocatore) è sufficiente, se il programma da rilocare ha un indirizzo iniziale maggiore o uguale a 49152.

In caso diverso (programma da rilocare

con indirizzo iniziale minore di 40960) occorrerà, come detto, una seconda istruzione che abbassi la cima della memoria.

Il programma da rilocare inizia alla locazione X, quindi la memoria BASIC dovrà essere abbassata fino alla locazione X-1.

Con procedura analoga calcoliamo i valori A e B da attribuire ai puntatori nelle locazioni 643 e 644:

$$B = \text{INT}((X-1)/256)$$

$$A = (X-1) - (B*256)$$

L'istruzione che abbassa la cima della memoria sarà quindi:

POKE643,A:POKE644,B:SYS58260

e andrà impartita subito dopo l'istruzione precedente, che alza l'inizio della memoria BASIC.

Si noti che l'utilizzo di queste due istruzioni, il cui effetto congiunto è quello di alzare l'inizio e abbassare la cima della memoria BASIC, porta logicamente a una diminuzione dello spazio BASIC di-

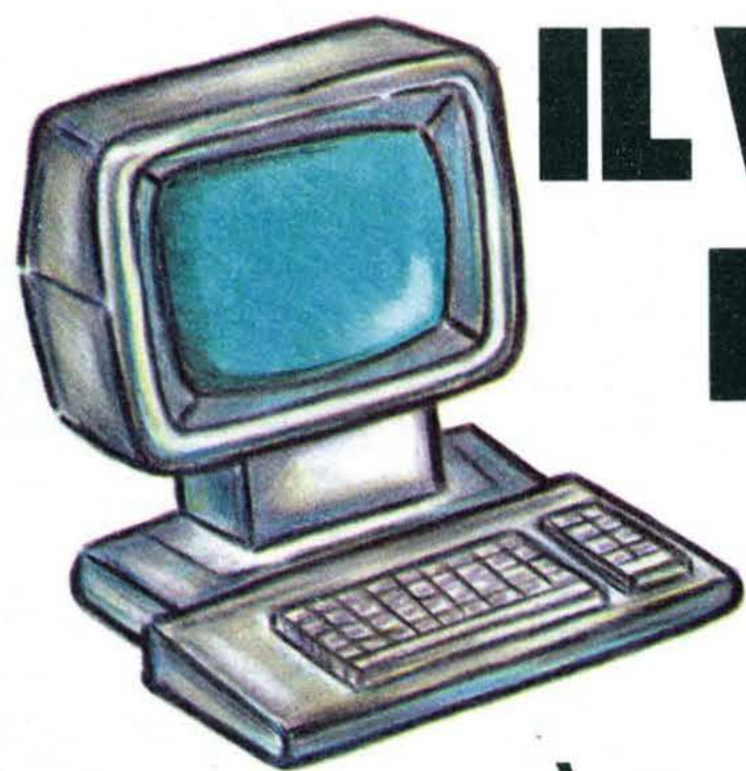
pendente dai valori X e Y.

Il Rilocatore sarà quindi in grado di rilocare correttamente quei programmi in linguaggio macchina la cui lunghezza non sia così eccessiva da non lasciare in memoria uno spazio sufficiente per ospitare il Rilocatore stesso e le sue variabili.

Memoria BASIC su misura

Attraverso questa tecnica è possibile riservare memoria per qualsiasi tipo di utilizzo: dall'immagazzinamento di dati per la costruzione di SPRITE a quello dei testi per un gioco di avventura (alcuni programmatori usano infatti "pokare" i testi in un'area di memoria protetta, in modo che dall'esame del listato del programma non risulti intelligibile la risoluzione dell'avventura).

Con un uso oculato di queste POKE è possibile avere in memoria più programmi BASIC contemporaneamente, anche se saranno eseguibili soltanto uno alla volta, oppure eseguire delle "fusioni" di più programmi in un unico e lungo file. Le possibili applicazioni sono molteplici, e permettono di ottenere risultati di tutto rispetto in sede di programmazione.



IL VOSTRO COMPUTER È BEN PROTETTO?



RICHIEDETE
DEPLIANTS E CARLO-
GHI INVIANDOCI QUESTO
COUPON
S.C.

COVER, LA CURA PIÙ EFFICACE PER LA PULIZIA E LA PROTEZIONE DEL COMPUTER



● Copertina protettiva in tessuto PVC.



● Copertura rigida in ABS per computer.



● Base porta stampanti.



● Borsa cofanetto per il trasporto del computer e accessori.



● Moduli ed etichette autoadesive in continuo.



Via L. Einaudi, 22
36040 BRENDOLA (VI)
Tel. 0444-798354
Telex 480824 I



● Cappa insonorizzante, per terminali stampanti.



● Prodotti per la pulizia del computer.



● Floppy disk "FARREL".

Uno sguardo alla grafica del C64

Quarta Parte

di J. Heilborn

trad. e adatt. di E. Stefani

Uno sguardo alla grafica del C64

Quarta parte

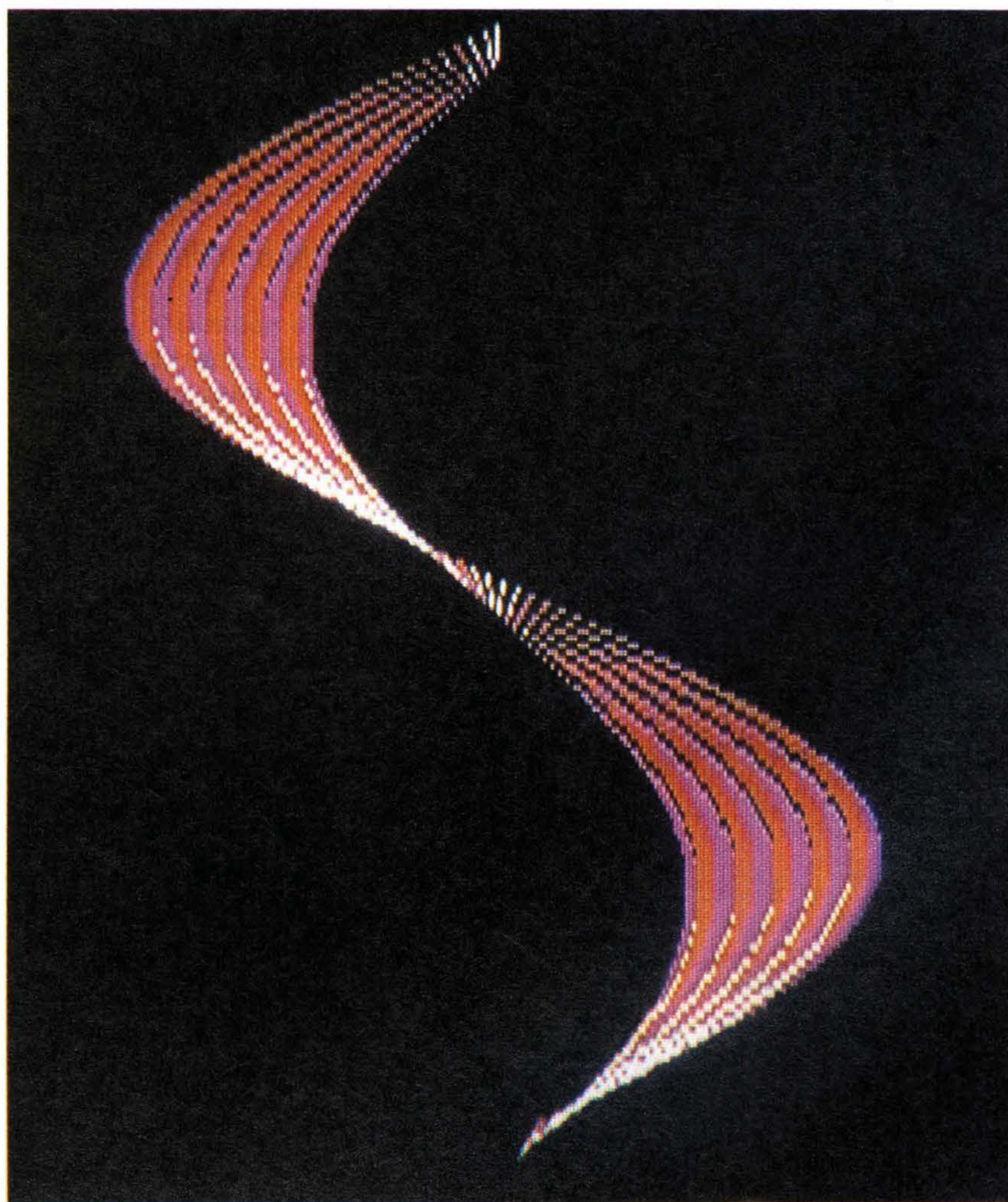
Dopo aver passato in rassegna, nelle precedenti puntate, le tecniche di programmazione dei caratteri e degli SPRITE, ci occupiamo questa volta di un argomento altrettanto affascinante: la realizzazione delle schermate in alta risoluzione.

Il terzo tipo di grafici che possono essere creati con il C64 è quello in alta risoluzione: lo schermo viene definito come un unico blocco di memoria, ed ogni bit del blocco viene visualizzato sul video come un singolo punto. Accendendo o spegnendo i singoli bit di questo blocco di memoria, si possono creare disegni plottati pixel per pixel.

Come lo schermo normale per la visualizzazione del testo, lo schermo in alta risoluzione è largo 40 caratteri per 25 righe in altezza. Tuttavia, nel modo in alta risoluzione (chiamato anche a mappa di bit) ogni posizione di un carattere non è semplicemente un singolo byte contenente il codice del carattere, bensì ogni locazione consiste di otto bit, che possono essere settati diversamente l'uno dall'altro. Il numero di celle nello schermo a mappa di bit è esattamente quello dello schermo normale: 1000.

Ogni cella può essere pensata come una griglia di 8 bit per 8, per un totale di 64 bit. Ciò significa che è possibile indirizzare singolarmente un totale di 64000 punti sullo schermo!

La differenza principale tra il modo ad al-



ta risoluzione e il modo testo normale è che nel secondo è possibile indirizzare singolarmente solo i pixel di ogni spazio-carattere (come nei caratteri programabili), mentre nel primo non si è assolutamente limitati dai singoli spazi-carattere, ma è come se si disponesse di un solo carattere grande quanto l'intero schermo.

L'organizzazione della memoria nel modo in alta risoluzione

I byte costituenti lo schermo in alta risoluzione sono organizzati nella stessa maniera del modo testo. Questo tipo di organizzazione rende molto facili da programmare linee di testo e qualche speciale applicazione grafica, mentre crea qualche difficoltà nella programmazione di grafici a mappa di bit.

Pensando lo schermo come una griglia, possiamo specificare un qualunque punto tramite le sue coordinate X ed Y. Immaginate una griglia divisa in otto qua-

drati per otto: per definire un singolo quadrato all'interno della griglia possiamo definire le sue coordinate X ed Y (che partono dal valore 0); ad esempio, il quadrato in posizione tre da sinistra e in posizione cinque dall'alto sarà definito dalla coppia di coordinate 2,4.

Possiamo accedere ai singoli punti entro ogni cella 8 per 8 dello schermo più o meno nel modo descritto sopra. Tuttavia, dal momento che lo schermo è organizzato con una sequenza di celle numerate per riga, dobbiamo innanzi tutto stabilire quale delle 1000 celle contiene il bit che vogliamo modificare. Solo allora potremo individuare il bit all'interno della cella.

Ricordate che lo schermo è suddiviso in 1000 celle di 8 per 8 bit, organizzate in 40 colonne per 25 righe. Cerchiamo ora di trovare la cella che contiene il punto a 55 pixel dal bordo sinistro dello schermo e a 43 pixel dal bordo superiore. Con il nostro sistema di coordinate X,Y l'origine è definita come il pixel nell'angolo superiore sinistro dello schermo, che avrà quindi coordinate 0,0. Il pixel alla sua destra avrà dunque coordinate 1,0, mentre quello subito sotto di esso sarà definito dalle coordinate 0,1.

Dal momento che stiamo cercando una cella 8 per 8, occorre dividere sia la coordinata orizzontale che quella verticale per otto, al fine di ottenere la posizione della cella (il resto della divisione viene per il momento ignorato). 55 diviso 8 è uguale a 6 (più un resto), quindi la coordinata X della cella è 6. 43 diviso 8 è uguale a 5 (più un resto), quindi la coordinata Y della cella è 5. Il bit 55,43 sarà dunque nella cella 6,5. Come si fanno questi calcoli in un programma? Questa linea pensa a tutto:

$CX=INT(BX/8):CY=INT(BY/8)$

Se BX e BY sono le coordinate X ed Y del bit, eseguita questa linea le variabili CX e CY conterranno rispettivamente le coordinate X ed Y della cella alla quale quel bit appartiene.

Come localizzare le celle di memoria e i bit relativi

Sfortunatamente non è sufficiente localizzare le celle nello schermo. Occorre anche trovare il byte ad esse corrispondente nella memoria del C64. Abbiamo visto che ogni cella consiste di otto byte, proprio come la configurazione di un carattere nel set. L'indirizzo di partenza di una cella (che è l'indirizzo del byte più alto della cella) è sempre esattamente superiore di 8 byte rispetto al-

l'indirizzo di partenza della cella immediatamente alla sua sinistra, ed è esattamente superiore di 320 byte rispetto all'indirizzo di partenza della cella immediatamente sopra di essa.

Se l'indirizzo di partenza della mappa di bit è immagazzinato nella variabile SB, l'indirizzo di partenza della cella 0,0 sarà SB+0, quello della cella 1,0 sarà SB+8, quello della 0,1 sarà SB+320, e così via. Così, per calcolare l'indirizzo di partenza di una cella della mappa di bit si userà la formula:

$CA=8*INT(BX/8)+320*INT(BY/8)$

dove BX e BY sono le coordinate del bit e CA è l'indirizzo di partenza, nella memoria del C64, della cella contenente il bit. Verifichiamolo con il bit 55,43 considerato nell'esempio precedente, e otterremo 1648. La cella contenente il bit di coordinate 55,43 partirà perciò dalla locazione di memoria SB+1648.

Abbiamo quindi calcolato l'indirizzo di partenza della cella. Ora dobbiamo però trovare quale sia il byte contenente il bit cercato all'interno della cella. Dal momento che i byte nelle celle sono disposti uno sopra l'altro, il byte cercato sarà sempre dato dal resto che abbiamo precedentemente trascurato dividendo per 8 la coordinata Y. Sfortunatamente il BASIC esprime sempre il resto di una divisione come numero frazionario: così, per ottenere da BASIC il resto come numero intero occorre sempre seguire questo procedimento:

$resto = numero - divisore * INT(numero / divisore)$

Ritornando al bit 55,43 dell'esempio, avremo che 43 diviso per 8 dà come risultato 5; il resto si otterrà moltiplicando 5 per 8 (=40) e sottraendolo dal numero originale: 43-40=3. Il resto sarà dunque 3. Ovviamente, disponiamo di un computer e sarà lui a dover eseguire tutti questi calcoli. Vediamo come.

La variabile CB conterrà l'indirizzo del byte cercato. La variabile R definisce il resto, mentre la variabile CV definisce la posizione verticale della cella:

$CV=INT(BY/8):R=BY-8*CV:CA=8*INT(BX/8)+320*CV:CB=CA+R$

In una sola linea vengono fatti parecchi calcoli: esaminiamo passo-passo il procedimento. Per prima cosa si calcola CV, la posizione verticale della cella, prendendo la parte intera di BY/8. Quindi si calcola R, il resto, sottraendo da BY il numero 8*CV. L'indirizzo di parten-

za della cella, CA, si calcola come abbiamo fatto precedentemente, con la differenza che ora CV equivale al risultato dell'operazione INT(BY/8). Infine si aggiunge R a CA per ottenere l'indirizzo del byte contenente il nostro bit. Prendendo il bit 55,43 dell'esempio, si otterrà come indirizzo del byte SB+1651.

La variabile CB contiene ora l'indirizzo del byte (naturalmente relativo a SB). Dobbiamo adesso trovare, all'interno del byte, il bit cercato. Fortunatamente, possiamo sfruttare semplicemente il resto della divisione BX/8 per trovare la posizione del bit. Usiamo la variabile BT per contenere la posizione del bit e la variabile CH come valore orizzontale temporaneo (nello stesso modo in cui abbiamo usato CV come valore verticale temporaneo):

$CV=INT(BY/8):R=BY-8*CV:CH=INT(BX/8):BT=BX-8*CH:CA=8*CH+320*CV:CB=CA+R$

La parte sottolineata è quella che provvede a calcolare BT, la posizione del bit. Si noti che CB, essendo usata per trovare l'indirizzo, è sempre la stessa per qualunque bit si voglia cercare nel byte. Ora che abbiamo calcolato la posizione del bit, vediamo come sfruttarla. La posizione del bit è un numero da 0 a 7 che rappresenta i bit in un byte nell'ordine seguente:

Posizione dei bit: 0 1 2 3 4 5 6 7

Usualmente tuttavia i bit vengono numerati in senso inverso:

Numerazione usuale: 7 6 5 4 3 2 1 0

C'è una buona ragione per questo cambiamento: se il bit 7 (numerazione usuale) è acceso, ha un valore di 128. Questo numero può essere espresso come potenza di 2 ($2^7=128$). Allo stesso modo il bit 6 vale 64 ($2^6=64$).

Il nostro procedimento di calcolo della posizione dei bit ci fornisce dei numeri che sono in ordine opposto a quello usuale. Fortunatamente essi possono essere facilmente convertiti nell'ordine usuale, sottraendo da 7 il numero trovato; a questo scopo basta porre $BT=7-BT$, e il valore che vogliamo usare dovrà essere espresso da 2^BT .

Attivare i pixel

Ora siamo pronti per disegnare un punto sullo schermo. Se lo schermo è pulito, possiamo farlo facilmente:

$POKESB+CB,2^BT$

Tuttavia, se esistono già sullo schermo dei punti che non desideriamo modificare, occorre stare attenti: dobbiamo attivare il bit nel byte di indirizzo SB+CB senza però modificare gli altri bit contenuti nello stesso byte. Infatti, potrebbe essere che il bit che stiamo accendendo sia già acceso: se si stanno tracciando linee, ad esempio, ciò può accadere molto frequentemente - dove una nuova linea ne interseca una già tracciata precedentemente - e il programma cercherà di accendere un bit già acceso. Per evitare questi inconvenienti accenderemo i bit usando l'operatore logico OR. Ricordiamo che l'operatore OR ha effetto sui bit solo se questi sono già accesi. Il comando per accendere i bit avrà quindi questa forma:

```
POKESB+CB,PEEK(SB+CB)OR2^BT
```

L'effetto di questa istruzione sarà quello di andare a vedere se il bit è già acceso e, se così fosse, lasciarlo inalterato.

Disattivare i pixel

Spegnere un bit è un procedimento del tutto simile, tranne che invece di porre il bit a 1 desideriamo porlo a 0. Useremo quindi l'operatore logico AND con il complemento di 2^{BT} . Il complemento di 2^{BT} è il numero binario definito da un 1 in tutte le posizioni del byte, tranne in quella in cui vogliamo disattivare il bit. Troviamo il complemento sottraendo da 255 il valore di 2^{BT} . Per spegnere un bit senza disturbare il resto del byte è sufficiente usare questa istruzione:

```
POKESB+CB,PEEK(SB+CB)AND255-2^BT
```

Un modo più veloce di operare sui singoli bit è quello di evitare l'elevamento a potenza (^) e la sottrazione da 255 e da 7, ponendo durante l'esecuzione del programma i valori corretti dei bit in appositi vettori. Useremo il vettore NG(n) per contenere i valori da usare per spegnere i bit e il vettore PS(n) per contenere i valori da usare per accendere i bit:

```
10 DIMNG(7),PS(7):GOSUB5000
5000 FORI=0TO7:BT=7-
      I:PS(I)=2^BT:NEXT
5010 FORI=0TO7:NG(I)=255-
      PS(I):NEXT:RETURN
```

Una volta inseriti nel programma questi vettori un bit può essere acceso più semplicemente con:

```
POKESB+CB,PEEK(SB+CB)ORPS(BT)
```

E può essere spento con:

```
POKESB+CB,PEEK(SB+CB)ANDNG(BT)
```

Per velocizzare ulteriormente le operazioni è possibile sostituire l'espressione SB+CB con una singola variabile che chiameremo AD:

```
AD=SB+CB:POKEAD,PEEK(AD)ORPS(BT)
```

Ecco ora un programma che comprende tutte le operazioni finora esaminate. Il programma vi chiederà le coordinate X,Y del punto sullo schermo ($0 \leq X \leq 319$, $0 \leq Y \leq 199$), quindi mostrerà l'indirizzo relativo del byte e i valori da inserire nella POKE per disegnare o cancellare il punto.

```
10 DIMNG(7),PS(7):GOSUB5000
100 PRINT"X (ORIZZONTALE)";:IN
    PUTBX
110 IFBX<0THENBX=0
120 IFBX>319THENBX=319
130 PRINT"Y (VERTICALE)";:
    INPUTBY
140 IFBY<0THENBY=0
150 IFBY>199THENBY=199
160 GOSUB200
170 PRINT"INDIRIZZO=SB+";CB
180 PRINT"DISEGNA:POKEAD,PEEK
    (AD)OR";PS(BT)
190 PRINT"CANCELLA:POKEAD,
    PEEK(AD)AND";NG(BT):
    PRINT:GOTO100
200 CV=INT(BY/8):R=BY-8*CV:
    CH=INT(BX/8):BT=BX-8*CH:
    CA=8*CH+320*CV:CB=CA+R
210 RETURN
5000 FORI=0TO7:BT=7-
    I:PS(I)=2^BT:NEXT
5010 FORI=0TO7:NG(I)=255-
    PS(I):NEXT:RETURN
```

Attivare il modo in alta risoluzione

Per far passare il C64 dal modo testo al modo in alta risoluzione è sufficiente una singola POKE:

```
POKE53265,PEEK(53265)OR32
```

Dopo questa istruzione si vedrà lo schermo riempirsi di caratteri strani e di vari colori, senza alcun senso apparente.

Normalmente il C64 prende le informazioni da visualizzare sul video da tre parti della memoria: la memoria di schermo (usualmente nelle locazioni da 1024 a 2023), il generatore di caratteri su ROM e la memoria colore, che controlla il colore dei caratteri e del fondo. Se si passa al modo in alta risoluzione, le informazioni per la gestione del video vengono raccolte da sezioni diverse di memoria. Ciò che viene visualizzato in alta risoluzione è contenuto in un blocco di memoria di 8000 byte, detto mappa di bit. I colori vengono invece presi dalla memoria di schermo (usualmente nelle locazioni 1024-2023). Modificando i valori imma-

gazzinati nella memoria di schermo, è possibile cambiare i colori visualizzati in alta risoluzione per creare un insieme più omogeneo:

```
FORR=1024TO2023:POKER,
230:NEXT
```

Dopo aver eseguito questa linea il nostro schermo diventerà il vecchio e familiare schermo blu con bordo azzurro (con un messaggio multicolore: il READY). Per maggiori informazioni sul modo di funzionare dei caratteri multicolori vedi le precedenti puntate. Per il momento diamo un'altra occhiata a ciò che si vede sullo schermo: il set caratteri è visualizzato nella sua metà inferiore e nella metà superiore sono visualizzati alcuni punti tremolanti. Cosa significa?

Ciò che stiamo vedendo sono i primi 8000 byte della memoria del C64. L'angolo superiore sinistro dello schermo visualizza la locazione 0, mentre l'angolo inferiore destro visualizza la locazione 7999. I byte tra 4096 e 7999 visualizzano buona parte del set caratteri. Ciò avviene perché, malgrado il set caratteri sia allocato tra 53248 e 57343, il chip VIC-II non potrebbe usarlo se fosse al di fuori del banco da 16 Kbyte che sta indirizzando attualmente. Per avviare a ciò il sistema operativo del C64 provvede a fare in modo che il chip VIC-II veda il set caratteri come allocato tra 4096 e 8191 mentre sta indirizzando il banco 0, che è il banco di 16 Kbyte normalmente indirizzato. I punti nella parte alta dello schermo sono parte del sistema operativo del computer, mentre quelli tremolanti sono dei bit nella memoria che vengono posti alternativamente a 1 e a 0 dal sistema operativo per i suoi scopi di gestione del computer. In un certo senso, stiamo guardando i "pensieri" del C64. Una parte dello schermo visualizza delle barrette verticali e parallele: rappresentano le locazioni da 1024 a 2023, che contengono il valore 230 prima immagazzinato con una POKE.

Pulire lo schermo ad alta risoluzione

Indagare sui misteri della visualizzazione di parti di memoria RAM o ROM tramite il VIC-II è molto interessante, ma non ci serve certo per creare disegni grafici in alta risoluzione. Per fare ciò, invece, occorre per prima cosa pulire lo schermo. Naturalmente, non è sufficiente premere come al solito i tasti SHIFT e CLR/HOME, in quanto questa funzione è operativa solo con lo schermo normale, in modo testo. Provate ugualmente a premere questi tasti, e vedrete lo scher-

mo passare al nero su rosso senza per altro cancellarsi. Ciò avviene perchè la memoria di schermo, normalmente cancellata inserendo nelle sue locazioni il valore 32 (codice dello spazio), controlla ora solo il colore, e quindi solo quest'ultimo viene modificato, lasciando inalterata la mappa di bit.

Per pulire lo schermo occorre inserire degli 0 negli 8000 byte usati per la mappa di bit. Tuttavia, attualmente la mappa di bit risiede in una zona parzialmente occupata dal sistema operativo, e questa operazione provocherebbe il blocco del computer. Occorre quindi spostare la mappa di bit in un'altra zona di memoria. Dal momento che lo spazio indirizzabile dal chip VIC-II è lungo 16 Kbyte, e la metà di questo spazio ci serve per la mappa di bit, abbiamo una sola possibilità: usare la seconda metà di questo blocco, da 8192 a 16384.

Muovere lo schermo ad alta risoluzione

Per muovere lo schermo ad alta risoluzione bisogna modificare il terzo bit del registro 54272 nel chip VIC-II. Se questo bit è 0, la mappa di bit si troverà nella metà inferiore del blocco video, mentre se questo bit è 1, la mappa di bit si troverà nella sua metà superiore. Ecco una routine che fa passare il C64 nel modo in alta risoluzione, muove la mappa di bit nella metà superiore del blocco di memoria da 16 Kbyte e la pulisce:

```
10 POKE53265,PEEK(53265)OR32
20 POKE53272,PEEK(53272)OR8
30 FORR=8192TO16191
40 POKER,0:NEXT
50 FORR=1024TO2023
60 POKER,230:NEXT
```

Eseguendo questa routine, vedrete inizialmente lo schermo riempirsi con blocchi colorati. Quindi questi saranno sostituiti dal familiare fondo blu. Alla fine dell'esecuzione della routine apparirà sullo schermo un gruppo di blocchi colorati: è il messaggio READY contenuto nella memoria di schermo, la quale ora gestisce i colori. Come evitarlo?

Al solito, è sufficiente mantenere il programma in ciclo, in modo che non venga mai visualizzato il messaggio READY. Aggiungete alla routine questa linea:

```
70 GOTO70
```

Combinando la routine precedente con un'altra simile a quella usata per calcolare le coordinate X ed Y di un singolo punto, siamo in grado di creare i primi disegni in alta risoluzione sul C64. Questa routine consente di disegnare in alta risoluzione usando i tasti cursore. Verrà

attivato e pulito automaticamente lo schermo in alta risoluzione - ci vogliono circa 20 secondi - quindi il computer aspetterà che venga premuto uno dei tasti cursore.

```
10 PRINT"{CLR}{BLU}";:X=
160:Y=100
20 FORR=1024TO2023:POKER,
230:NEXT
30 POKE53272,PEEK(53272)OR8:
POKE53265,PEEK(53265)OR32
40 FORR=8192TO16383:POKER,0:
NEXT
50 GETA$:IFA$=""THEN50
60 IFA$="{SU}"THENY=Y-1:
GOTO110
70 IFA$="{GIU}"THENY=Y+1:
GOTO110
80 IFA$="{DES}"THENX=X+1:
GOTO110
90 IFA$="{SIN}"THENX=X-1:
GOTO110
100 GOTO50
110 P=(8192+(INT(Y/8))*320+
(INT(X/8))*8+(YAND7))
120 POKEP,PEEK(P)ORINT(2^(7-
((INT(X)/8)-INT
T(INT(X)/8))*8))
130 GOTO50
```

La linea 10 cancella lo schermo e pone il punto di partenza del plottaggio al centro dello schermo (X=160 e Y=100). La linea 20 pone il colore di sfondo a blu scuro e quello delle linee plottate ad azzurro (per maggiori dettagli vedi le puntate precedenti). La linea 30 entra in modo ad alta risoluzione e muove la mappa di bit nella metà superiore del blocco di memoria da 16 Kbyte.

La linea 40 pulisce la mappa di bit. La linea 50 legge il tasto premuto e quindi aggiorna il valore delle coordinate.

Le linee 110 e 120 accendono il punto sullo schermo e la linea 130 rimanda alla linea 50 di lettura del tasto premuto.

I grafici in alta risoluzione

L'alta risoluzione permette di avere un controllo bit per bit di tutto lo schermo, e non solo di un singolo carattere o di un singolo SPRITE. Ciò rende possibile creare disegni estremamente elaborati, ma rende più complesse le procedure da seguire per ottenerli. Un carattere è composto da 64 bit, uno SPRITE da 504 bit, lo schermo in alta risoluzione è composto da 64000 bit e richiede quindi la disposizione di un intero blocco di memoria di 8000 byte: la metà esatta di quanto possa gestire il chip VIC-II.

I colori in alta risoluzione

Come viene gestito il colore nello schermo ad alta risoluzione? Lo schermo trae le forme visualizzate dalla mappa di bit, un blocco di memoria di 8000 byte che dobbiamo definire quando entriamo nel modo ad alta risoluzione. Ciò libera migliaia di byte della memoria di schermo

che inizia alla locazione 1024. Questa zona di memoria viene ora usata per controllare individualmente i colori della mappa di bit. La solita memoria colore in 55296-56319 non viene usata. Ogni byte della memoria di schermo corrisponde a una cella (blocco di 8 byte) della mappa di bit. Al contrario della memoria colore, che dispone solo dei nybble bassi (bit da 0 a 3), la memoria di schermo possiede 8 bit per ogni byte. Dal momento che occorrono solo 4 bit per contenere il valore del colore (da 0000 a 1111, o da 0 a 15 in decimale), ogni locazione della memoria di schermo può controllare due colori. E' esattamente così che il C64 gestisce i colori in modo ad alta risoluzione.

Il nybble alto (i quattro bit di sinistra) di ogni locazione controlla il colore di primo piano delle celle (cioè quello dei bit posti a 1), mentre il nybble basso controlla il colore di fondo di ogni cella (quello dei bit posti a 0). In altre parole, per ogni cella della mappa di bit esiste un byte che controlla sia il colore di primo piano che quello di fondo della cella. Se il valore del byte è 11010110 (decimale 214), il nybble alto 1101 controlla il colore di primo piano, mentre il nybble basso 0110 controlla quello di fondo. In questo caso i colori di primo piano e di fondo sarebbero rispettivamente 13 e 6: verde chiaro e blu.

Per mettere il valore corrispondente al verde chiaro 13 (binario 1101) in un nybble alto è sufficiente moltiplicarlo per 16. Per regolare una cella su un colore di primo piano verde chiaro e un colore di fondo verde, si userà questa istruzione:

POKESM,5+16*3

La formula generale è:

POKE memoria di schermo, fondo
+16*primo piano

Per vedere questi concetti tradotti in pratica ecco una routine che crea uno schermo a mappa di bit diviso in tre parti. La parte alta consiste in punti alternativamente accesi e spenti, che misceleranno il colore di primo piano con quello di fondo. La parte centrale è formata da punti spenti, in modo che vi sia visualizzato solo il colore di fondo. La parte bassa consiste di punti tutti accesi che mostrano il colore di primo piano.

```
10 POKE53272,29:POKE53265,59:
POKE56576,198:A=1
20 FORR=17408TO18407POKER,A:
NEXT
30 FORR=24576TO27135STEP2
40 POKER,85:POKER+1,170:NEXT
50 FORR=27136TO30015:
```



```

POKER,0: NEXT
60 FORR=30016TO32575:
   POKER,255:NEXT
70 FORM=0TO255
80 FORR=17408TO18407:POKER,M:
   NEXT
90 NEXT

```

Il programma visualizzerà tutte le 256 combinazioni di colori. Potete vedere come i vari colori si miscelano sullo schermo.

Il modo multicolore

Mentre è possibile creare nuovi diversi colori con la tecnica appena vista della miscelazione, non è tuttavia possibile avere più di due colori in un singolo carattere o in una singola cella. Ciò significa che, mentre la risoluzione dei nostri disegni può arrivare al singolo bit o pixel, la risoluzione del colore arriva al massimo a otto bit per otto: non certo un gran risultato!

Il modo multicolore, disponibile sia in modo testo che in modo ad alta risoluzione, risolve il problema, permettendo di avere fino a quattro colori per ogni carattere o cella. C'è un prezzo da pagare, tuttavia: la risoluzione grafica si dimezza in senso orizzontale (un punto sarà quindi largo due pixel), mentre rimane invariata in senso verticale. I caratteri standard, una volta abilitato il modo multicolore, saranno per questo motivo molto poco leggibili; se si desiderano quindi disegni precisi e testo ben leggibile, è da preferirsi il modo standard. Invece, per certi set di caratteri ridefiniti o per schermate molto colorate, il modo multicolore testo e il modo multicolore in alta risoluzione forniranno un'alternativa eccellente.

Sia nel modo testo che nel modo in alta risoluzione, i caratteri od i disegni sono stati formati con l'uso di singoli bit accesi o spenti: un 1 rappresenta un punto nel colore di primo piano, e uno 0 rappresenta un punto nel colore di sfondo. L'usare un solo bit per definire un punto permette solo due possibili colori contemporaneamente. Il modo multicolore, invece,

non usa un singolo bit per una istruzione di colore, ma ne usa una coppia. Nel modo standard un singolo byte comprende otto istruzioni di colore acceso-spento.

Nel modo multicolore ogni byte comprende solo quattro istruzioni di colore, ma ognuna di esse è lunga due bit. Pensate ad ogni byte come composto da quattro coppie: 10 10 10 10. Ogni coppia di bit ha quattro possibili valori: 00, 01, 10 e 11. Una volta attivato il modo multicolore il chip VIC-II legge i byte come gruppi di quattro coppie di bit invece che come semplici otto bit.

In ogni caso, l'uso delle coppie di bit crea un altro problema: ci sono ora solo quattro istruzioni colore per byte invece di otto. Abbiamo raddoppiato il numero dei colori disponibili, ma dimezzato il numero di punti che possono essere controllati singolarmente. Ogni coppia di bit in una certa configurazione controlla ora due pixel invece di uno soltanto, mentre ogni byte controlla ancora otto pixel sullo schermo. Ciò significa che la risoluzione orizzontale è stata dimezzata, mentre quella verticale è rimasta inalterata. Un punto sullo schermo sarà dunque alto un pixel e largo due.

Il modo multicolore a mappa di bit è abilitato attivando sia il modo a mappa di bit che il modo multicolore:

```

POKE53265,PEEK(53265)OR32:POKE
53270,PEEK(53270)OR16

```

La mappa di bit è formata da celle di 8 byte, esattamente come nel modo in alta risoluzione. Ogni byte della mappa è trattato come insieme di quattro coppie di bit, ognuna delle quali controlla un doppio punto sullo schermo, come nel modo testo multicolore. Riassumiamo infine i due passi fondamentali per l'utilizzo del modo multicolore ad alta risoluzione:

* *Assegnare i colori:* in questo caso le assegnazioni dei colori sono un po' più complesse. Il colore di fondo (coppie di bit 00) per l'intero schermo della mappa

di bit è tratto dalla locazione 53281: il colore di fondo non può essere assegnato individualmente per ogni cella.

I colori di primo piano (coppie di bit 11) vengono letti dalla memoria colore (55296-56319): tutti e 16 i colori sono disponibili in ogni locazione e il modo multicolore è abilitato per ogni cella.

Gli altri due colori sono assegnati individualmente nella memoria di schermo, un byte per ogni cella della mappa di bit. Il nybble alto controlla le coppie di bit 01, mentre il nybble basso controlla le coppie di bit 10.

* *Regolare le coppie di bit all'interno della mappa:* vi accorgete rapidamente che le routine usate per disegnare in modo standard ad alta risoluzione danno risultati alquanto strani in modo multicolore a mappa di bit. La risoluzione orizzontale è ora dimezzata e le linee (in particolare le diagonali o le linee curve) saranno molto meno precise e definite, e cambieranno anche i colori visualizzati. Il sistema migliore per imparare a programmare in questi modi grafici è sperimentare. Probabilmente i modi grafici di uso più complesso sono quelli a mappa di bit, dal momento che richiedono una notevole manipolazione di numeri binari. I modi testo sono molto più semplici da programmare e da usare. Disponendo di un editor di caratteri, troverete facile creare complessi disegni: potete ridefinire interi set di caratteri e diversi modi testo, quindi imparare ad usarli direttamente da tastiera.

Con questo abbiamo concluso il nostro ideale viaggio all'interno del complesso mondo dei principali modi grafici del C64. Alcune tecniche di programmazione, come in particolare queste relative al modo in alta risoluzione a mappa di bit, non sono certamente facili né di immediata e perfetta comprensione per chi non sia un buon programmatore; consigliamo quindi di sperimentare quanto suggerito in queste puntate, per passare delle ore divertenti imparando con il proprio computer.

WORK

Autodisk

di M. Cristuib Grizzi

Ecco un programma per automatizzare il caricamento dei propri file su disco e per eliminare il problema di dover ricordare la SYS di attivazione dei programmi in linguaggio macchina.

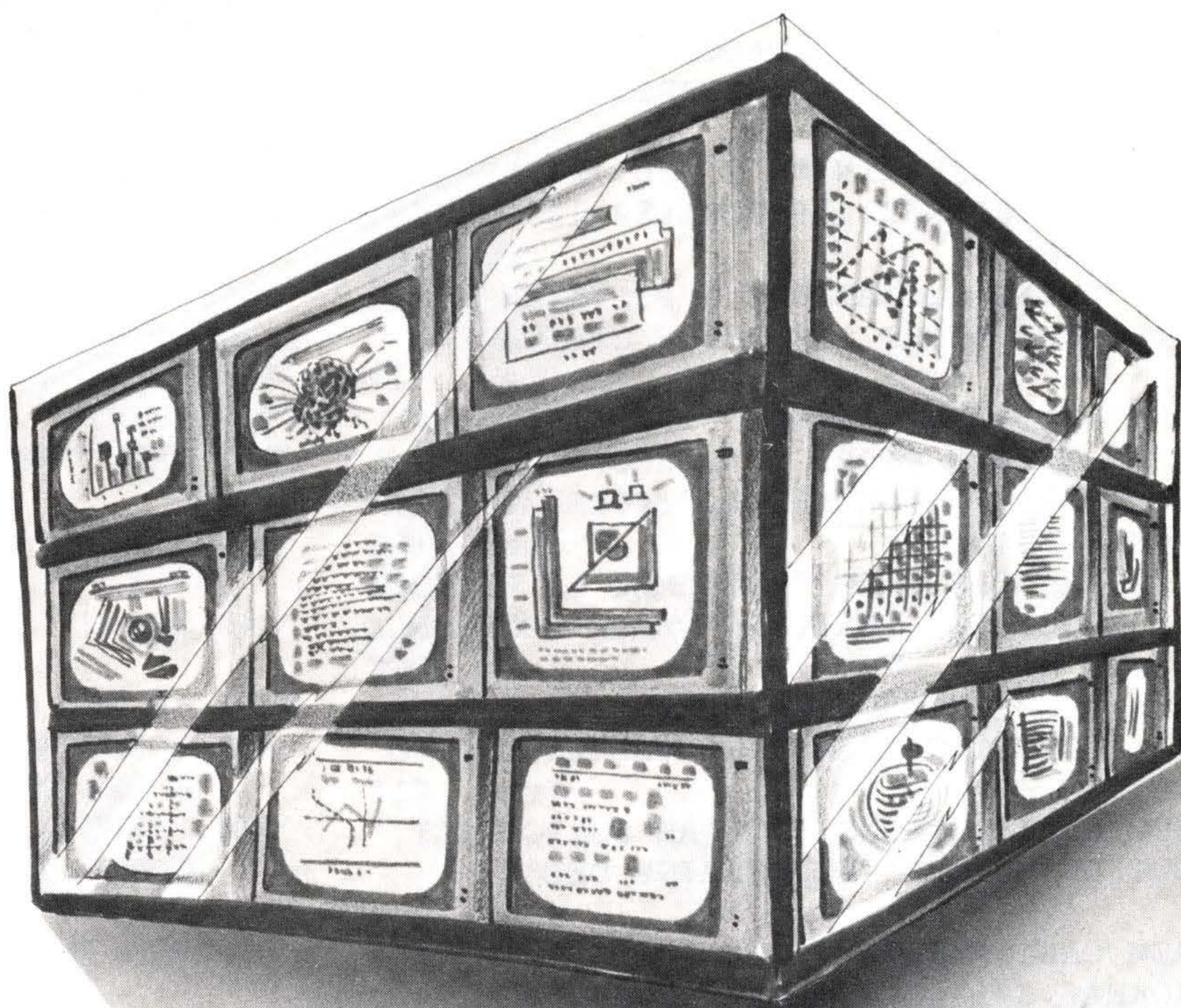
Autodisk è un programma che, una volta salvato su ciascuno dei vostri dischetti, vi permetterà di disporre di un comodo menu di selezione, comprendente tutti i file presenti sul dischetto nel drive, attraverso il quale potrete caricarli e lanciarli automaticamente.

Dopo aver attivato Autodisk ci sarà una breve attesa, necessaria a leggere il contenuto della directory del disco e a inserirne il contenuto in un apposito vettore; verrà quindi elencato sul video il contenuto del disco, con ciascun file contraddistinto da un numero progressivo.

Sarà allora sufficiente premere il tasto corrispondente al numero del programma che si desidera caricare, per fare in modo che Autodisk ne legga l'indirizzo iniziale e lo carichi in memoria, attivandolo automaticamente.

Se l'indirizzo iniziale del file è diverso da 2049 (inizio dell'area BASIC), Autodisk provvede ad eseguire la SYS di attivazione, seguita da un'istruzione NEW, mentre se l'indirizzo di caricamento è proprio 2049, verrà impartito un semplice RUN.

Se non si desidera caricare nessuno dei programmi presenti sul dischetto, sarà sufficiente premere RETURN alla domanda di Autodisk, per passare così all'esame di un nuovo disco.



Alcune precauzioni

La tecnica dell'"overlay" di programmi BASIC nei computer Commodore richiede che il programma caricatore debba essere più lungo del programma caricato: in caso contrario il secondo programma viene caricato soltanto parzialmente.

Autodisk "inganna" il computer attraverso la linea 230, che sposta i puntatori in

modo da creare una falsa occupazione di memoria, maggiore di quella effettivamente richiesta dal programma. Il valore 64 delle tre istruzioni POKE di questa linea significa che la massima lunghezza dei programmi caricabili da Autodisk è di circa 64 blocchi su disco.

Questo valore è più che sufficiente nella maggior parte dei casi: se tuttavia si desiderasse caricare file più lunghi, è possibile modificare questi tre valori

con il massimo numero di blocchi da caricare. Si tenga tuttavia presente che deve rimanere lo spazio in memoria per le variabili e i vettori sia di Autodisk che del programma caricato, pena un messaggio di "OUT OF MEMORY" al termine del caricamento.

A causa di questa "finta" occupazione di memoria, è opportuno *non ris salvare mai* su disco un programma caricato da

Autodisk, in quanto esso verrebbe salvato insieme a tutta la memoria fino alla locazione 16384 ($64 \times 256 = 16384$) o maggiore, se si modificano i valori in linea 230. Si utilizzi quindi Autodisk soltanto per caricare e attivare i programmi su disco che non richiedono ulteriori salvataggi.

Autodisk non è inoltre compatibile con quei programmi che modificano i punta-

tori 55 e 56 (ad esempio Shifter su questo numero della rivista), nè con i programmi in linguaggio macchina che hanno una SYS di attivazione diversa dall'indirizzo iniziale (ad esempio Ultrawedge).

Naturalmente, Autodisk non può caricare file sequenziali e relativi, ma soltanto file programma.

Autodisk

```

100 REM AUTODISK - MCG 1986 :rem 181
110 REM UTILIZZARE QUESTA UTILITY SOLTANT
    O PER CARICARE I PROGRAMMI :rem 28
120 REM NON RISALVARE MAI PROGRAMMI CARIC
    ATI ATTRAVERSO AUTODISK! :rem 96
130 REM ROUTINE DI AUTOSTART PER I PROGRA
    MMI IN LINGUAGGIO MACCHINA :rem 129
140 POKE198,0:IFH+L<>0THENPRINT"{CLR}NEW"
    :rem 113
150 IFH+L<>0THENPRINT"{ 2 GIU'}SYS"H+L:PO
    KE198,3:POKE631,19:POKE632,13:POKE633
    ,13:END :rem 171
160 REM INIZIALIZZAZIONE E POSIZIONAMENTO
    DEI PUNTATORI PER POTER CARICARE
    :rem 21
170 REM PROGRAMMI LUNGHI FINO A 16 KBYTE
    (CIRCA 64 BLOCCHI SU DISCO) :rem 111
180 POKE53280,0:POKE53281,0:POKE53272,21:
    POKE646,1:CLR :rem 106
190 REM PER CARICARE PROGRAMMI DI LUNGHEZ
    ZA MAGGIORE, INSERIRE NELLE POKE DELL
    A :rem 107
200 REM LINEA SEGUENTE IL MASSIMO VALORE
    DI BLOCCHI CARICABILE. VALORI TROPPO
    :rem 213
210 REM ALTI POSSONO CAUSARE SPIACEVOLI '
    OUT OF MEMORY' AL TERMINE DEL
    :rem 239
220 REM CARICAMENTO. :rem 204
230 POKE46,64:POKE72,64:POKE50,64:CLR
    :rem 171
240 PRINT"{CLR}{RVS}{CYN}{ 6 SPAZI}A
    { 2 SPAZI}U{ 2 SPAZI}T{ 2 SPAZI}O
    { 2 SPAZI}D{ 2 SPAZI}I{ 2 SPAZI}S
    { 2 SPAZI}K{ 6 SPAZI}MCG 86{WHT}"
    :rem 89
250 C$=CHR$(13):P$=CHR$(34) :rem 149
260 N$="" :rem 139
270 PRINT"{ 7 GIU'}{ 8 SPAZI}CARICAMENTO
    DIRECTORY" :rem 188
280 PRINT"{ 6 GIU'}{ 12 SPAZI}ATTENDERE..
    ." :rem 247
290 DIM F$(155) :rem 203
300 FOR J=0 TO 49 :rem 65
310 F$(J)="" :rem 26
320 NEXT :rem 212
330 J=0 :rem 77
340 REM APERTURA FILE SU DIRECTORY DISCO
    :rem 206
350 OPEN 8,8,0,"$0" :rem 134
360 GOSUB750 :rem 181
370 PRINT"{ 6 SU}{ 17 SPAZI}PER":rem 184
380 PRINT"{GIU'}{ 11 SPAZI}{RVS}";F$(0);"
    {OFF}" :rem 198
390 J=1 :rem 84
400 GOSUB750 :rem 176
410 F$(1)="" :rem 2
420 EF=0 :rem 142
430 GOSUB750 :rem 179
440 IF EF=0 THEN J=J+1:GOTO430 :rem 133
450 T=J-1 :rem 210
460 CLOSE8 :rem 72
470 PRINT"{CLR}" :rem 255
480 J=1:EF=0 :rem 134
490 GOSUB830 :rem 184
500 IF EF=0 GOTO490 :rem 242
510 PRINT :rem 35
520 PRINT"RETURN PER ALTRO DISCO, O"
    :rem 158
530 PRINT"{GIU'}CARICA PROGRAMMA ( DA 1 A
    ";T;"); :rem 84
540 INPUT"{ 4 SPAZI}{ 3 SIN}";A$:N=VAL(A$
    ) :rem 194
550 IF N>0 AND N<=T GOTO610 :rem 163
560 PRINT"{CLR}{ 2 GIU'}INSERISCI IL DISC
    O NEL DRIVE E" :rem 111
570 PRINT"{ 2 GIU'}PREMI RETURN SE SEI PR
    ONTO{ 6 SPAZI}{ 5 SIN}";:INPUTA$
    :rem 130
580 REM INIZIALIZZAZIONE DEL DRIVE
    :rem 163
590 OPEN15,8,15:PRINT#15,"I0" :rem 101
600 RUN :rem 139
610 PRINT"{CLR}{ 10 GIU'}{ 3 DES}CARICAME
    NTO DI{ 2 SPAZI}{RVS}";F$(N);"{OFF}"
    :rem 22
620 REM LETTURA INDIRIZZO INIZIALE DEL FI
    LE SU DISCO :rem 195
630 OPEN1,8,3,"0:"+F$(N) :rem 205
640 GET #1,L$ :rem 106
650 GET #1,H$ :rem 103
660 X$=H$:GOSUB720:H=X*256 :rem 242
670 X$=L$:GOSUB720:L=X :rem 52
680 PRINT"{ 2 GIU'}{ 3 SPAZI}INDIRIZZO IN
    IZIALE:"H+L :rem 161
690 CLOSE1 :rem 70
700 REM CARICAMENTO DEL FILE IN RAM
    :rem 13
710 LOAD F$(N),8,1 :rem 130
720 IF X$="" THEN X=0:RETURN :rem 51
730 X=ASC(X$) :rem 211
740 RETURN :rem 123

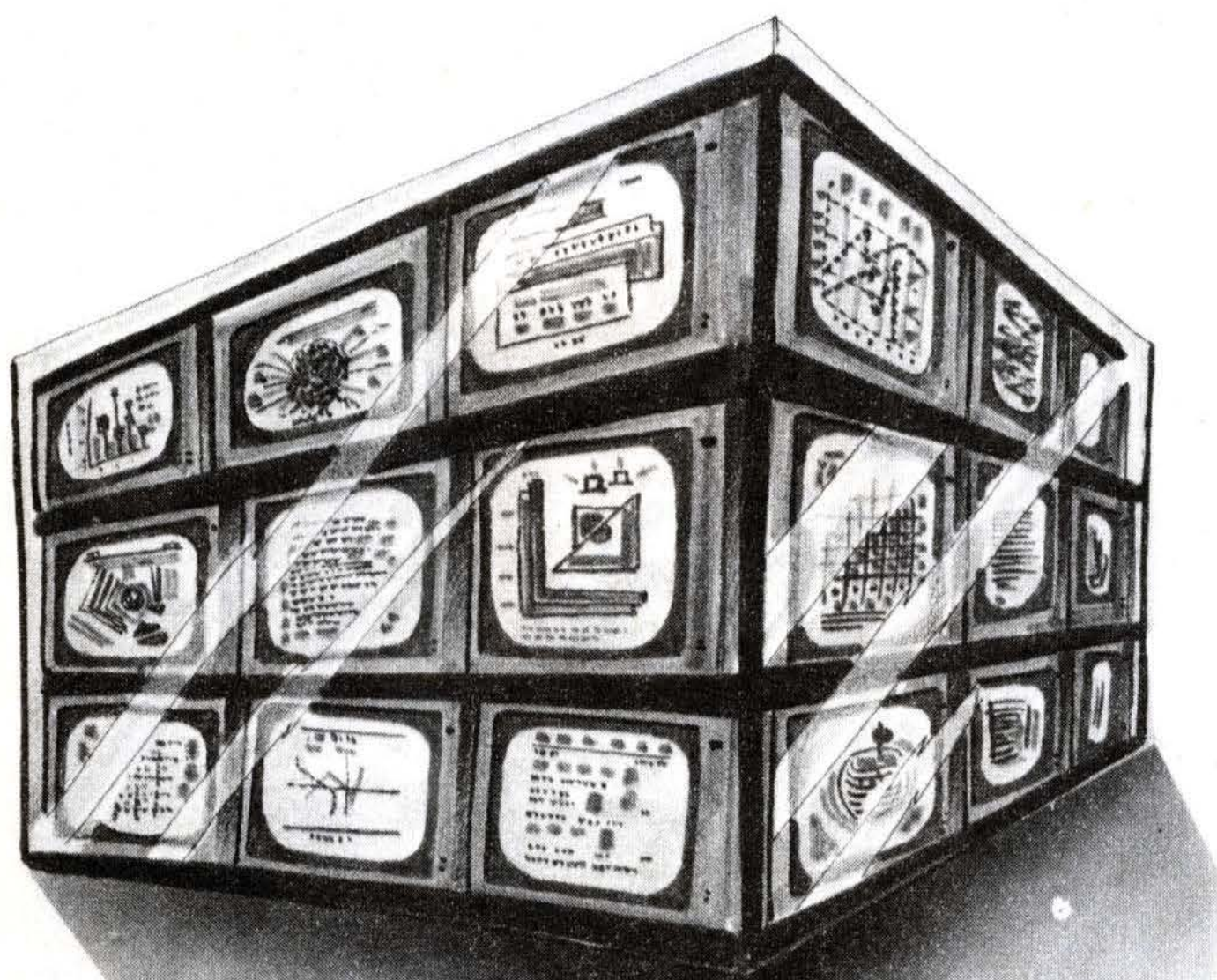
```



```

750 GET#8,A$:IFA$<>N$THEN750      :rem 77
760 GET#8,A$:IFA$<>N$THEN790      :rem 82
770 GET#8,A$:IFA$<>N$THEN790      :rem 83
780 GET#8,A$:IFA$=N$THENEFF=1:RETURN
                                     :rem 138
790 IFA$<>P$THEN760                :rem 78
800 GET#8,A$:IFA$=P$THEN820        :rem 12
810 F$(J)=F$(J)+A$:GOTO800         :rem 123
820 RETURN                          :rem 122
830 JA$=STR$(J):JB$=STR$(J+1)      :rem 122
840 PRINT "{RVS}";RIGHT$(JA$,2);"{OFF}";TAB(3);F$(J);
                                     :rem 220
850 IF (J=35) OR (J>75) OR (J=115) THENGOSUB910
                                     :rem 171
860 IF F$(J+1)="" THEN PRINT:GOTO880
                                     :rem 222
870 PRINTTAB(20);"{RVS}";RIGHT$(JB$,2);"{OFF}";TAB(23);F$(J+1)
                                     :rem 248
880 IF J>T-2 THEN EF=1              :rem 146
890 J=J+2                          :rem 207
900 RETURN                          :rem 121
910 PRINT:PRINT "{GIU' } {RVS} C {OFF} ONTINUA
O {RVS} S {OFF} CEGLI?"            :rem 49
920 GETAA$:IFAA$="C" THENPRINT "{CLR}":RETU
RN                                  :rem 1
930 IFAA$="S" THEN510               :rem 106
940 GOTO920                         :rem 113

```



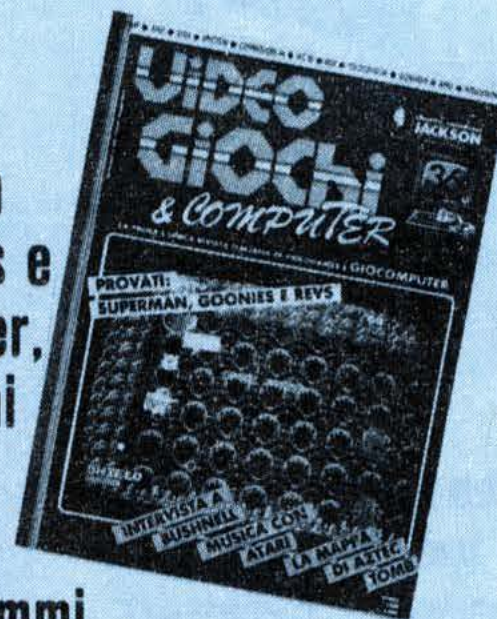
JACKSON

per il tuo "computertempo" libero

Audiotest, rassegne,
computer music,
servizi, interviste
e uno speciale inserto
"Bit-Musica"
per chi suona
col proprio personal.



La più bella
e ricca rivista
di videogames e
home computer,
con recensioni
di giochi
sempre nuovi
e tanti programmi
per Atari, Commodore,
Sharp, Sinclair,
Texas ...



I giochi più
entusiasmanti
per C64
importati
dall'Inghilterra
mai presentati
in Italia.
Una sfida Jackson
al già visto,
al già fatto,
al ...già registrato.



La rivista
per gli utenti
di Commodore
64 e 128.
Giochi, utilities
e notizie.

... tienile
vicino
al tuo
joystick!

In busta chiusa inviate questo coupon a:
Gruppo Editoriale Jackson
20124 Mi - Via Rosellini, 12

☐ Desidero ricevere GRATIS un numero
della Rivista _____

(allego L. 1.000 in francobolli per
contributo spese di spedizione)

☐ Desidero essere informato sulle
prossime iniziative del
Gruppo Editoriale Jackson.

Nome _____
Cognome _____
Professione _____
via _____ Tel. _____
CAP _____ Città _____



**GRUPPO EDITORIALE
JACKSON**
Milano-San Francisco-Londra-Madrid

WORK

Screen genie

di J. A. Ledger
trad. e adatt. di L. Priotto

Questa utility grafica vi permetterà di disegnare facilmente intere schermate con caratteri grafici: quando tutto è pronto Screen Genie scrive automaticamente un programma BASIC in grado di ricreare la videata grafica. Il programma utilizza come memoria di massa sia il drive che il registratore.

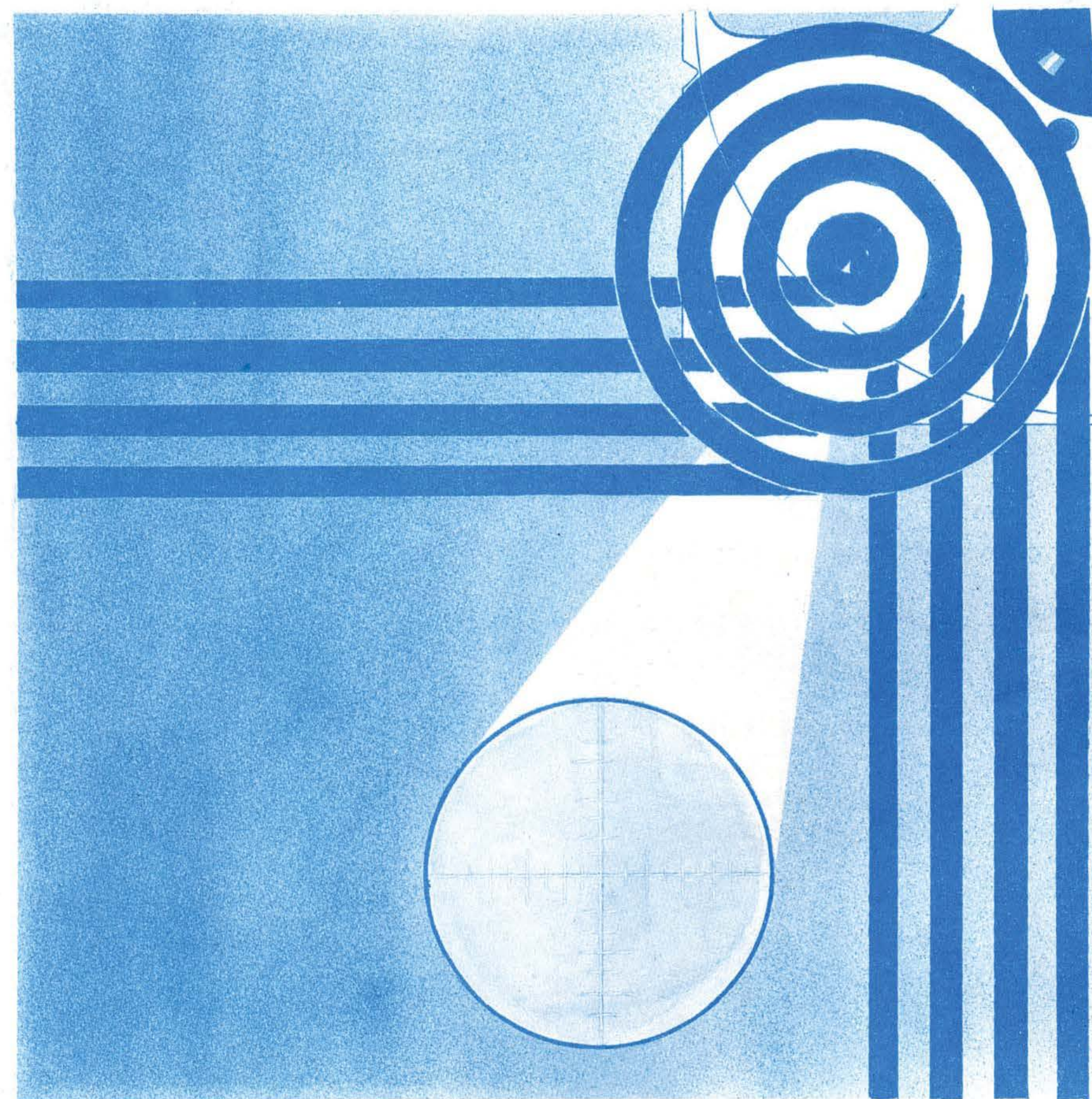
Screen Genie può essere definito sia un editor di schermate che un generatore di programmi: attraverso di esso potete facilmente creare sfondi per i vostri giochi, videate coloratissime per titoli, o più semplicemente pagine di istruzioni.

Con Screen Genie si può disegnare utilizzando caratteri di testo oppure caratteri grafici, in modo normale, multicolore oppure a colore di fondo esteso.

Anche se Screen Genie si carica e attiva come un qualsiasi programma BASIC, dopo il RUN esso si divide in vari moduli, che vanno ad allocarsi completamente al di fuori della memoria BASIC, lasciandovi il massimo dello spazio per i programmi.

Per prima cosa viene visualizzata una schermata di aiuto contenente i principali comandi (questa schermata è sempre richiamabile premendo CTRL+H). La selezione di un comando qualsiasi nella schermata di aiuto provoca il passaggio allo schermo di lavoro e l'esecuzione del comando stesso.

Una volta entrati nello schermo di lavoro quasi tutti i tasti producono il loro effetto normale e consueto: sia il testo che i



caratteri grafici possono essere visualizzati normalmente, con eccezione del punto interrogativo e del tasto INST/DEL, che non può essere utilizzato insieme allo SHIFT per inserire un carattere.

Il tasto INST/DEL (senza lo SHIFT) non è disabilitato, ma funziona in maniera leggermente differente da quella consueta: cancella infatti il carattere che si trova nella posizione del cursore e spo-

sta quest'ultimo di uno spazio verso sinistra, senza far scorrere la parte di testo che si trova a destra del carattere cancellato.

Inoltre, per evitare scorrimenti dello schermo, non è possibile digitare alcunché nell'angolo inferiore destro dello schermo stesso: questo spazio viene utilizzato dal programma per mostrare il colore selezionato per i caratteri che si stanno digitando.

I comandi di Screen Genie si dividono in quattro gruppi: comandi immediati, modi di schermo, funzioni del cursore, controllo del colore.

Comandi immediati

I comandi immediati sono i seguenti:

CTRL+H: visualizza la schermata di aiuto.

CTRL+T: cancella lo schermo dall'angolo superiore sinistro alla posizione in cui si trova il cursore.

CTRL+B: cancella lo schermo dall'angolo inferiore destro alla posizione in cui si trova il cursore.

CTRL+M: muove un blocco di caratteri da una posizione all'altra sullo schermo. Prima di poter muovere un blocco occorre definirne gli angoli superiore sinistro e inferiore destro: premete CTRL+M, posizionate il cursore in corrispondenza dell'angolo superiore sinistro del blocco e premete RETURN. Spostate quindi il cursore sull'angolo inferiore destro del blocco di caratteri e premete nuovamente RETURN: ora il blocco è definito. Non resta che posizionare il cursore nel punto in cui desiderate sia trasferito l'angolo superiore sinistro del blocco e premere RETURN: il contenuto della nuova area sarà sostituito dal blocco di caratteri (si noti che l'area originale non viene disturbata).

Questo comando non permette di spostare un blocco di caratteri in un'area che si espanderebbe oltre il bordo dello schermo.

CTRL+Z: memorizza la schermata in un buffer situato nella RAM del computer. Una schermata così memorizzata può essere richiamata in qualsiasi momento premendo CTRL+O.

CTRL+O: sostituisce la schermata visualizzata con quella contenuta nel buffer, e viceversa. Questo comando ha anche una funzione cosiddetta di "undo", in quanto permette di annullare anche gli effetti dei comandi CTRL+M e di cancellazione schermo.

CTRL+P: scrive una sequenza di linee BASIC in grado di ricreare la schermata visualizzata. Queste linee, che iniziano con un numero di vostra scelta, possono essere "fuse" con qualsiasi programma BASIC si trovi in memoria.

Se si verificano conflitti tra i numeri di linea del programma generato da Screen Genie e quelli del programma in memo-

ria, viene visualizzato un messaggio che consente di scegliere un nuovo numero di linea iniziale.

E' inoltre possibile istruire Screen Genie in modo che aggiunga anche delle linee relative al colore di bordo e fondo e, se ci si trova in multicolore o in modo a colore di fondo esteso, per far sì che aggiunga anche le linee che permettono di accedere a questi modi grafici.

Infine avete la possibilità di aggiungere una linea che attenda la pressione di un tasto da parte dell'utente, linea particolarmente utile se si utilizzano più videate. Se selezionate questa opzione mentre vi trovate in modo multicolore oppure a colore di fondo esteso, potete anche fare in modo che venga aggiunta una linea in grado di riattivare il normale modo testo dopo che sia avvenuta la pressione di un tasto.

La risposta "di default" per tutte le opzioni di questo modo programma è affermativa (Y): la pressione di qualsiasi tasto che non sia Y oppure RETURN equivale a una risposta negativa.

CTRL+X: provoca l'uscita in ambiente BASIC. Per riattivare Screen Genie è sufficiente digitare GENIE e premere RETURN.

Si noti che l'istruzione GENIE funziona sia in modo diretto che in modo programma.

Modi di schermo

Oltre al normale modo testo, Screen Genie permette di utilizzare il modo multicolore e il modo a colore di fondo esteso. Per un'analisi del funzionamento di questi modi grafici fate riferimento alla serie di articoli "Uno sguardo alla grafica del C64" che stiamo pubblicando. I comandi per i modi grafici di schermo sono:

CTRL+K: entra nel modo a colore di fondo esteso, in cui è possibile utilizzare soltanto i primi 64 caratteri del set.

CTRL+C: entra in modo multicolore. Quest'ultimo non può essere utilizzato insieme al modo a colore di fondo esteso, e offre i massimi risultati se utilizzato con un set di caratteri ridefiniti.

CTRL+U: permette l'uso di caratteri ridefiniti. In questo modo grafico sono disponibili soltanto i caratteri maiuscoli e grafici. Prima di usare questo comando dovete caricare in memoria il set di caratteri ridefinito, a partire dalla locazione 61440. Si noti che se si utilizza un set di caratteri ridefiniti, al programma BASIC eventualmente generato da

Screen Genie occorrerà aggiungere la serie di istruzioni necessarie al caricamento e abilitazione del nuovo set di caratteri.

Funzioni del cursore

Corrispondono ai tasti funzione di numero dispari:

F1: permette di disegnare con un carattere. Esistono due modi per selezionare il carattere da utilizzare: muovere il cursore sopra il carattere desiderato e premere F1, oppure premere F1 e digitare il carattere che si intende utilizzare. Per cancellare premete la barra spazio.

F3: seleziona il colore in cui si disegna: premete CTRL oppure il tasto Commodore insieme a un tasto numerico da 1 a 8.

F5: modifica l'aspetto di un carattere, facendolo passare da maiuscolo a minuscolo e viceversa. E' molto utile in modo a colore di fondo esteso, dove i caratteri shiftati hanno un diverso colore di fondo.

F7: pone in campo inverso un carattere. Anche questo comando è molto utile in modo a colore di fondo esteso, dove l'inversione di un carattere gli fa assumere un diverso colore di fondo.

Controllo del colore

I tasti funzione di numero pari presiedono invece al controllo del colore:

F2: modifica il colore del bordo dello schermo.

F4: modifica il colore del fondo dello schermo.

F6: permette di modificare il colore dei registri (in modo multicolore o colore di fondo esteso) uno, due e tre.

F8: modifica il colore di tutti i caratteri colorati come il carattere che si trova in corrispondenza del cursore.

Note per il caricamento, la duplicazione su disco e la digitazione del programma

Se si dispone della cassetta allegata alla rivista, il programma si carica in memoria e si manda in esecuzione attraverso la pressione contemporanea dei tasti SHIFT e RUN/STOP. La cassetta contiene, nello stesso ordine in cui appaiono sulla rivista, tutti i programmi di questo numero: per ottenerne un elenco, con la

relativa posizione del contanastro, inserire nel registratore la cassetta completamente riavvolta, azzerare il contanastro e digitare:

VERIFY ""

Poichè la cassetta sicuramente non contiene alcun programma dal nome "" verranno passati in rassegna tutti i programmi contenuti attraverso il messaggio:

FOUND nome del programma

Risulta utile annotarsi i nomi dei vari programmi e la relativa posizione del contanastro (sottrarre due o tre unità dal valore corrispondente all'apparizione del messaggio FOUND), in modo da poter in seguito caricare il programma desiderato senza dover necessariamente legge-

re tutti i programmi precedenti.

Se si desidera duplicare il programma su disco, è sufficiente caricarlo in memoria da cassetta con un semplice LOAD (senza cioè mandarlo in esecuzione) e quindi salvarlo su disco con il comando SAVE"nome programma",8.

Gli abbonati che non dispongono della cassetta dovranno invece digitare il programma attraverso MLX (in altra parte della rivista). Poichè il linguaggio macchina da digitare occupa l'area di memoria (inizio del BASIC) che sarebbe normalmente occupata da MLX stesso, occorre riconfigurare la memoria del C64 prima di caricare MLX. A questo scopo, si esegua la seguente procedura:

1. Spegner e riaccendere il C64 e digitare in modo diretto (senza numero di linea) le seguenti istruzioni:

POKE44,32:POKE32*256,0:NEW

2. Caricare e mandare in esecuzione MLX, rispondendo alle sue due domande iniziali con:

indirizzo iniziale: 2049

indirizzo finale: 7440

3. Digitare il programma in linguaggio macchina e salvarlo su nastro o disco.

Una volta salvato il programma su nastro esso potrà essere ricaricato in memoria in qualsiasi momento, senza più bisogno di MLX, seguendo le istruzioni date più sopra per i possessori di cassetta.

Si ricorda che se il programma è stato salvato su disco, occorre aggiungere il nome del file e il numero di periferica 8 all'istruzione LOAD.

Screen Genie

```
2049 :024,008,010,000,158,050,251
2055 :048,055,052,058,083,067,114
2061 :082,069,069,078,032,071,158
2067 :069,078,073,069,000,000,052
2073 :000,169,004,133,014,169,002
2079 :013,133,015,169,000,133,238
2085 :016,169,160,133,017,162,182
2091 :015,160,000,136,177,014,033
2097 :145,016,192,000,208,247,089
2103 :230,015,230,017,202,048,029
2109 :006,208,238,160,136,208,249
2115 :234,160,128,185,139,028,173
2121 :153,111,198,136,208,247,102
2127 :160,000,152,153,000,199,231
2133 :136,208,250,169,255,141,220
2139 :021,199,160,063,153,064,239
2145 :199,136,016,250,160,021,111
2151 :169,192,153,128,199,136,056
2157 :136,136,016,248,169,255,045
2163 :141,128,199,141,131,199,030
2169 :141,210,199,141,213,199,200
2175 :160,015,169,003,153,192,051
2181 :199,136,136,136,016,248,236
2187 :120,169,034,133,001,169,253
2193 :208,160,224,162,016,032,179
2199 :239,170,169,038,133,001,133
2205 :088,032,120,171,024,032,112
2211 :146,164,169,001,032,207,114
2217 :170,169,000,173,017,208,138
2223 :072,041,239,141,017,208,125
2229 :169,009,162,009,032,155,205
2235 :165,056,032,146,164,104,086
2241 :133,072,032,201,170,169,202
2247 :023,162,010,032,155,165,234
2253 :169,200,160,184,162,004,060
2259 :032,239,170,169,216,160,173
```

```
2265 :188,162,004,032,239,170,244
2271 :169,225,162,009,032,155,207
2277 :165,169,000,141,001,008,201
2283 :141,002,008,169,003,133,179
2289 :045,133,047,133,049,169,049
2295 :008,133,046,133,048,133,236
2301 :050,173,017,208,009,016,214
2307 :141,017,208,076,057,164,154
2313 :147,013,013,032,032,028,018
2319 :018,002,008,032,042,042,159
2325 :042,032,087,079,082,075,162
2331 :045,083,067,082,069,069,186
2337 :078,032,042,042,042,002,015
2343 :008,032,146,002,005,013,245
2349 :002,004,032,030,084,079,020
2355 :032,082,069,045,069,078,170
2361 :084,069,082,032,065,070,203
2367 :084,069,082,032,069,088,231
2373 :073,084,032,084,079,032,197
2379 :066,065,083,073,067,013,186
2385 :013,002,004,032,084,089,049
2391 :080,069,032,018,032,071,133
2397 :069,078,073,069,032,146,048
2403 :032,065,078,068,032,080,198
2409 :082,069,083,083,032,082,024
2415 :069,084,085,082,078,013,010
2421 :013,002,004,032,152,040,104
2427 :032,079,082,032,018,032,142
2433 :083,089,083,032,053,048,005
2439 :056,048,048,032,146,032,241
2445 :041,002,006,013,002,004,209
2451 :032,129,080,082,069,083,110
2457 :083,032,018,083,072,073,002
2463 :070,084,045,067,076,082,071
2469 :146,032,084,079,032,067,093
2475 :076,069,065,082,032,083,066
```


2481 :067,082,069,069,078,002,032
 2487 :007,013,002,004,032,031,016
 2493 :067,085,082,082,069,078,140
 2499 :084,032,067,079,076,079,100
 2505 :082,032,073,078,068,073,095
 2511 :067,065,084,069,068,032,080
 2517 :072,069,082,069,045,045,083
 2523 :045,045,045,062,019,000,179
 2529 :152,019,029,029,192,192,070
 2535 :192,192,192,192,192,192,103
 2541 :192,192,192,192,192,192,109
 2547 :192,192,192,192,192,192,115
 2553 :192,192,192,192,192,192,121
 2559 :192,192,192,192,002,004,005
 2565 :017,002,008,157,029,029,247
 2571 :029,029,029,029,029,029,185
 2577 :029,029,029,029,029,000,162
 2583 :147,158,117,002,038,099,072
 2589 :105,098,032,032,117,099,000
 2595 :105,117,099,105,176,099,224
 2601 :105,117,099,105,117,099,171
 2607 :105,178,032,178,032,032,092
 2613 :117,099,105,117,099,105,183
 2619 :178,032,178,032,178,117,006
 2625 :099,105,032,032,098,098,017
 2631 :032,032,106,099,105,098,031
 2637 :032,032,098,099,107,171,104
 2643 :179,032,171,179,032,098,006
 2649 :109,098,032,032,098,099,045
 2655 :178,171,179,032,098,109,094
 2661 :098,032,098,171,179,032,199
 2667 :032,032,098,098,032,032,175
 2673 :106,096,107,106,099,107,222
 2679 :177,032,109,106,099,107,237
 2685 :106,099,107,177,032,177,055
 2691 :032,032,106,099,107,106,101
 2697 :099,107,177,032,177,032,249
 2703 :177,106,099,107,032,032,184
 2709 :098,106,002,038,099,107,087
 2715 :153,117,002,018,096,105,134
 2721 :156,117,002,018,096,105,143
 2727 :153,098,018,067,079,077,147
 2733 :077,065,078,068,083,058,090
 2739 :146,032,032,067,084,082,110
 2745 :076,032,038,032,098,156,105
 2751 :098,018,032,067,085,082,061
 2757 :083,079,082,032,070,085,116
 2763 :078,067,084,073,079,078,150
 2769 :083,032,146,098,153,098,051
 2775 :002,018,032,098,156,098,107
 2781 :002,018,032,098,153,098,110
 2787 :072,045,072,069,076,080,129
 2793 :032,040,084,072,073,083,105
 2799 :032,077,069,078,085,041,109
 2805 :098,156,098,070,049,045,249
 2811 :068,082,065,087,002,010,053
 2817 :032,095,098,153,098,066,031
 2823 :045,067,076,069,065,082,155
 2829 :032,084,079,032,066,079,129

2835 :084,084,079,077,032,098,217
 2841 :156,098,070,051,045,080,013
 2847 :065,073,078,084,002,009,086
 2853 :032,095,098,153,098,084,085
 2859 :045,067,076,069,065,082,191
 2865 :032,084,079,032,084,079,183
 2871 :080,002,004,032,098,156,171
 2877 :098,070,053,045,067,072,210
 2883 :065,078,071,069,032,067,193
 2889 :065,083,069,032,032,032,130
 2895 :095,098,153,098,077,045,133
 2901 :077,079,086,069,002,012,154
 2907 :032,098,156,098,070,055,088
 2913 :045,082,069,086,069,082,018
 2919 :083,069,002,007,032,095,135
 2925 :098,153,098,090,045,077,158
 2931 :069,077,079,082,073,090,073
 2937 :069,032,083,067,082,069,011
 2943 :069,078,032,098,156,106,154
 2949 :002,018,096,107,153,098,095
 2955 :079,045,079,079,080,083,072
 2961 :033,032,040,082,069,083,228
 2967 :084,079,082,069,041,032,026
 2973 :098,154,117,002,018,096,130
 2979 :105,153,098,080,045,067,199
 2985 :082,069,065,084,069,032,058
 2991 :080,082,073,078,084,032,092
 2997 :080,071,077,098,154,098,247
 3003 :018,002,006,032,067,079,135
 3009 :076,079,082,002,007,032,215
 3015 :146,098,153,098,088,045,059
 3021 :069,088,073,084,032,084,123
 3027 :079,032,066,065,083,073,097
 3033 :067,032,032,032,098,154,120
 3039 :098,002,018,032,098,153,112
 3045 :106,002,018,096,107,154,200
 3051 :098,070,050,045,066,079,131
 3057 :082,068,069,082,002,009,041
 3063 :032,098,150,117,002,018,152
 3069 :096,105,154,098,070,052,060
 3075 :045,066,065,067,075,071,136
 3081 :082,079,085,078,068,002,147
 3087 :005,032,098,150,098,018,160
 3093 :032,077,079,068,069,083,173
 3099 :058,032,146,032,067,084,190
 3105 :082,076,032,038,032,032,069
 3111 :032,098,154,098,070,054,033
 3117 :045,066,065,067,075,071,178
 3123 :078,068,032,082,069,071,195
 3129 :032,049,045,051,098,150,226
 3135 :098,002,018,032,098,154,209
 3141 :098,070,056,045,067,072,221
 3147 :065,082,083,032,079,070,230
 3153 :032,083,065,077,069,032,183
 3159 :032,098,150,098,075,045,073
 3165 :069,088,084,069,078,068,037
 3171 :069,068,032,066,065,067,210
 3177 :075,071,068,095,098,154,154
 3183 :098,032,032,032,067,079,195

3189 :076,079,082,032,065,083,022
 3195 :032,067,072,065,082,032,217
 3201 :032,098,150,098,067,045,107
 3207 :077,085,076,084,073,067,085
 3213 :079,076,079,082,002,005,208
 3219 :032,095,098,154,098,032,144
 3225 :032,032,079,086,069,082,021
 3231 :032,067,085,082,083,079,075
 3237 :082,002,004,032,098,150,021
 3243 :098,085,045,085,083,069,124
 3249 :082,032,068,069,070,073,059
 3255 :078,069,068,032,032,032,238
 3261 :095,098,154,106,002,018,150
 3267 :096,107,150,098,032,032,198
 3273 :067,072,065,082,032,083,090
 3279 :069,084,032,064,032,054,030
 3285 :049,052,052,048,098,152,152
 3291 :032,032,080,082,069,083,085
 3297 :083,032,065,078,089,032,092
 3303 :075,069,089,013,150,106,221
 3309 :002,018,096,107,152,032,132
 3315 :032,070,079,082,032,087,113
 3321 :079,082,075,045,083,067,168
 3327 :082,069,069,078,000,173,214
 3333 :134,002,141,044,208,032,054
 3339 :195,165,036,085,016,025,021
 3345 :160,018,217,055,175,240,114
 3351 :005,136,016,248,048,013,233
 3357 :152,010,168,185,074,175,025
 3363 :072,200,185,074,175,072,045
 3369 :096,201,148,240,214,201,117
 3375 :020,208,010,169,149,162,253
 3381 :172,032,155,165,076,000,141
 3387 :160,201,034,240,196,201,067
 3393 :147,208,006,024,032,146,116
 3399 :164,169,147,072,032,008,151
 3405 :171,104,134,041,132,042,189
 3411 :224,024,208,012,201,013,253
 3417 :240,169,201,141,240,165,221
 3423 :201,017,240,161,036,012,250
 3429 :080,033,201,157,240,029,073
 3435 :201,029,240,025,201,145,180
 3441 :240,021,201,017,240,017,081
 3447 :032,172,160,166,041,164,086
 3453 :042,024,032,240,255,032,238
 3459 :195,161,076,154,160,032,141
 3465 :172,160,032,008,171,224,136
 3471 :024,208,012,192,039,208,058
 3477 :008,164,042,166,041,024,082
 3483 :032,240,255,032,048,166,160
 3489 :032,245,160,032,191,160,213
 3495 :032,220,160,032,232,160,235
 3501 :076,000,160,201,013,240,095
 3507 :004,201,141,208,008,232,205
 3513 :160,000,024,032,240,255,128
 3519 :096,076,210,255,036,013,109
 3525 :080,024,160,000,177,014,140
 3531 :073,128,145,014,041,063,155
 3537 :201,032,208,010,173,033,098

3543 :208,041,015,205,137,175,228
 3549 :240,007,096,036,013,016,117
 3555 :007,160,000,173,134,002,191
 3561 :145,016,096,036,012,016,042
 3567 :008,160,000,177,014,073,159
 3573 :064,145,014,096,036,012,100
 3579 :080,010,160,000,165,010,164
 3585 :145,014,165,011,145,016,241
 3591 :096,024,032,146,164,032,245
 3597 :048,166,169,231,133,016,008
 3603 :169,203,133,017,032,136,197
 3609 :166,076,000,160,024,032,227
 3615 :146,164,032,048,166,165,240
 3621 :014,133,016,165,015,133,001
 3627 :017,169,000,133,014,169,033
 3633 :200,133,015,032,136,166,219
 3639 :076,000,160,238,032,208,001
 3645 :076,000,160,238,033,208,008
 3651 :076,000,160,165,013,073,042
 3657 :128,133,013,032,048,166,081
 3663 :032,220,160,076,000,160,215
 3669 :165,013,073,064,133,013,034
 3675 :032,048,166,032,191,160,208
 3681 :076,000,160,024,032,146,023
 3687 :164,076,000,160,173,017,181
 3693 :208,073,064,141,017,208,052
 3699 :041,064,240,008,173,022,151
 3705 :208,041,239,141,022,208,212
 3711 :076,000,160,173,022,208,254
 3717 :073,016,141,022,208,041,122
 3723 :016,240,008,173,017,208,033
 3729 :041,191,141,017,208,076,051
 3735 :000,160,173,024,208,073,021
 3741 :004,041,252,141,024,208,059
 3747 :076,000,160,165,012,073,137
 3753 :128,133,012,032,048,166,176
 3759 :032,232,160,076,000,160,067
 3765 :165,012,073,064,133,012,128
 3771 :032,048,166,032,195,161,053
 3777 :032,245,160,076,000,160,098
 3783 :160,000,177,014,133,010,181
 3789 :177,016,133,011,096,056,182
 3795 :032,146,164,032,048,166,031
 3801 :160,000,177,014,041,191,032
 3807 :044,017,208,080,002,041,103
 3813 :063,201,032,208,003,076,044
 3819 :000,160,177,016,041,015,132
 3825 :133,074,169,000,133,014,252
 3831 :133,016,169,216,133,017,163
 3837 :169,252,133,015,160,000,214
 3843 :162,004,032,129,170,177,165
 3849 :014,032,135,170,041,015,160
 3855 :197,074,208,009,177,016,184
 3861 :024,105,001,041,015,145,096
 3867 :016,200,208,230,230,017,160
 3873 :230,015,202,208,223,032,175
 3879 :195,165,201,140,240,198,154
 3885 :076,009,160,056,032,146,012
 3891 :164,032,205,170,169,169,192

3897 :162,174,032,155,165,169,146
 3903 :049,133,016,141,249,174,057
 3909 :169,214,162,174,032,155,207
 3915 :165,230,016,165,016,201,100
 3921 :052,208,238,169,013,162,155
 3927 :175,032,155,165,169,028,043
 3933 :141,021,208,160,002,185,042
 3939 :067,000,153,041,208,136,192
 3945 :016,247,032,243,169,201,245
 3951 :032,208,012,169,000,141,161
 3957 :021,208,056,032,237,164,067
 3963 :076,000,160,201,049,144,241
 3969 :233,201,052,176,229,056,052
 3975 :233,048,168,182,066,232,040
 3981 :138,153,066,000,076,092,154
 3987 :162,024,032,146,164,032,195
 3993 :172,163,169,024,133,059,105
 3999 :169,039,133,060,169,030,247
 4005 :141,248,203,032,182,163,110
 4011 :134,041,134,057,132,042,199
 4017 :132,058,165,014,133,076,243
 4023 :165,015,133,077,165,016,242
 4029 :133,078,165,017,133,079,026
 4035 :173,000,208,141,012,208,169
 4041 :173,001,208,141,013,208,177
 4047 :173,016,208,041,191,141,209
 4053 :016,208,041,001,074,106,147
 4059 :106,013,016,208,141,016,207
 4065 :208,169,002,141,045,208,230
 4071 :141,046,208,169,096,141,008
 4077 :021,208,169,031,141,248,031
 4083 :203,032,182,163,138,056,249
 4089 :229,041,133,061,152,056,153
 4095 :229,042,133,062,169,024,146
 4101 :056,229,061,133,059,169,200
 4107 :039,056,229,062,133,060,078
 4113 :173,000,208,141,014,208,249
 4119 :173,001,208,141,015,208,001
 4125 :173,016,208,041,127,141,223
 4131 :016,208,041,001,074,106,225
 4137 :013,016,208,141,016,208,131
 4143 :169,224,141,021,208,166,208
 4149 :041,164,042,024,032,240,084
 4155 :255,169,030,141,248,203,081
 4161 :032,172,163,032,182,163,041
 4167 :165,014,133,080,165,015,131
 4173 :133,081,165,016,133,082,175
 4179 :165,017,133,083,165,076,210
 4185 :133,014,165,077,133,015,114
 4191 :169,000,133,016,169,248,062
 4197 :133,017,032,231,163,165,074
 4203 :078,133,014,165,079,133,197
 4209 :015,169,000,133,016,169,103
 4215 :252,133,017,032,231,163,179
 4221 :169,000,133,014,169,248,090
 4227 :133,015,165,080,133,016,161
 4233 :165,081,133,017,032,231,028
 4239 :163,169,000,133,014,169,023
 4245 :252,133,015,165,082,133,161

4251 :016,165,083,133,017,032,089
 4257 :231,163,169,032,141,021,150
 4263 :208,169,028,141,248,203,140
 4269 :076,000,160,169,000,133,199
 4275 :057,133,058,096,032,210,253
 4281 :255,032,195,165,072,032,168
 4287 :008,171,104,201,145,208,004
 4293 :006,228,057,208,237,240,149
 4299 :238,201,017,208,006,228,077
 4305 :059,208,227,240,228,201,092
 4311 :157,208,006,196,058,208,024
 4317 :217,240,218,201,029,208,054
 4323 :006,196,060,208,207,240,120
 4329 :208,096,166,061,032,129,157
 4335 :170,164,062,177,014,145,203
 4341 :016,169,032,145,014,136,245
 4347 :016,245,032,135,170,165,246
 4353 :014,024,105,040,133,014,075
 4359 :144,002,230,015,165,016,067
 4365 :024,105,040,133,016,144,219
 4371 :002,230,017,202,016,212,186
 4377 :096,024,032,237,164,076,142
 4383 :000,160,032,120,171,056,058
 4389 :032,146,164,032,201,170,014
 4395 :169,184,160,200,162,004,154
 4401 :032,239,170,169,188,160,239
 4407 :216,162,004,032,239,170,110
 4413 :165,012,041,192,010,170,139
 4419 :042,141,182,217,138,010,029
 4425 :042,141,102,217,165,013,241
 4431 :041,192,010,170,042,141,163
 4437 :142,217,138,010,042,141,007
 4443 :222,217,165,072,041,064,104
 4449 :010,010,042,141,050,219,057
 4455 :165,071,041,016,240,002,126
 4461 :169,001,141,090,219,165,126
 4467 :070,041,004,074,074,141,007
 4473 :130,219,169,000,133,198,202
 4479 :032,220,165,036,085,048,201
 4485 :007,056,032,237,164,076,193
 4491 :000,160,072,056,032,237,184
 4497 :164,104,076,009,160,176,066
 4503 :013,169,136,133,014,169,017
 4509 :175,133,015,169,176,072,129
 4515 :208,011,169,065,133,014,251
 4521 :169,000,133,015,169,248,135
 4527 :072,160,007,173,017,208,044
 4533 :145,014,136,173,022,208,111
 4539 :145,014,136,173,024,208,119
 4545 :145,014,136,185,032,208,145
 4551 :041,015,145,014,136,016,054
 4557 :246,173,134,002,133,073,198
 4563 :165,199,133,075,032,008,055
 4569 :171,134,041,132,042,104,073
 4575 :168,169,200,162,004,032,190
 4581 :239,170,164,017,169,216,180
 4587 :162,004,032,239,170,096,170
 4593 :008,176,017,056,032,146,164
 4599 :164,169,136,133,014,169,008

4605 :175,133,015,169,176,072,225
 4611 :208,028,169,065,133,014,108
 4617 :169,000,133,015,169,248,231
 4623 :072,165,075,133,199,165,056
 4629 :073,141,134,002,166,041,066
 4635 :164,042,024,032,240,255,016
 4641 :160,007,177,014,141,017,037
 4647 :208,136,177,014,141,022,225
 4653 :208,136,177,014,141,024,233
 4659 :208,136,177,014,153,032,003
 4665 :208,136,016,248,032,129,058
 4671 :170,160,200,162,004,104,095
 4677 :032,239,170,165,015,160,082
 4683 :216,162,004,032,239,170,130
 4689 :032,135,170,040,176,026,148
 4695 :032,129,170,169,248,160,227
 4701 :176,162,008,032,239,170,112
 4707 :032,135,170,160,007,185,020
 4713 :065,000,153,136,175,136,002
 4719 :016,247,169,032,141,021,225
 4725 :208,096,160,000,166,016,251
 4731 :208,002,198,017,198,016,250
 4737 :166,014,208,002,198,015,220
 4743 :202,134,014,032,049,172,226
 4749 :177,014,032,057,172,145,226
 4755 :016,228,095,208,225,166,061
 4761 :015,228,096,208,219,096,247
 4767 :133,014,134,015,160,000,103
 4773 :177,014,240,014,201,002,045
 4779 :240,011,032,210,255,200,095
 4785 :208,242,230,015,208,238,038
 4791 :096,200,177,014,170,200,016
 4797 :177,014,032,210,255,202,055
 4803 :208,250,240,233,032,048,182
 4809 :166,169,001,013,021,208,011
 4815 :141,021,208,173,039,208,229
 4821 :073,001,141,039,208,169,076
 4827 :020,133,019,133,018,032,062
 4833 :243,169,170,165,145,073,166
 4839 :251,208,006,169,128,133,102
 4845 :085,048,012,224,133,144,115
 4851 :004,224,141,144,242,169,143
 4857 :000,133,085,160,000,132,247
 4863 :018,169,254,045,021,208,202
 4869 :141,021,208,138,096,024,121
 4875 :032,146,164,162,034,189,226
 4881 :144,175,149,008,202,016,199
 4887 :248,162,034,189,179,175,242
 4893 :149,057,202,016,248,120,053
 4899 :162,214,160,175,024,032,034
 4905 :141,255,169,004,141,136,119
 4911 :002,088,076,118,198,169,186
 4917 :000,133,015,133,016,056,150
 4923 :032,240,255,152,201,040,211
 4929 :144,003,233,040,168,010,151
 4935 :010,010,038,016,105,024,018
 4941 :141,000,208,144,002,230,034
 4947 :016,173,016,208,041,254,023
 4953 :005,016,141,016,208,138,101

4959 :010,010,010,133,016,105,123
 4965 :050,141,001,208,165,016,170
 4971 :010,038,015,010,038,015,233
 4977 :024,101,016,144,002,230,118
 4983 :015,132,016,024,101,016,167
 4989 :133,014,133,016,169,200,022
 4995 :101,015,133,015,105,016,004
 5001 :133,017,096,160,000,169,200
 5007 :032,145,014,166,014,228,230
 5013 :016,208,006,166,015,228,020
 5019 :017,240,008,230,014,208,104
 5025 :238,230,015,208,234,096,158
 5031 :169,032,141,231,203,056,231
 5037 :032,146,164,032,205,170,154
 5043 :169,147,032,210,255,169,137
 5049 :000,133,020,133,021,133,113
 5055 :064,169,255,162,172,032,021
 5061 :155,165,169,021,162,175,020
 5067 :032,155,165,032,195,165,179
 5073 :076,232,166,056,032,237,240
 5079 :164,076,000,160,169,182,198
 5085 :162,172,032,155,165,076,215
 5091 :180,166,032,195,165,201,142
 5097 :013,240,067,201,032,240,002
 5103 :228,201,048,144,231,201,012
 5109 :058,176,227,032,210,255,179
 5115 :041,015,168,165,020,166,058
 5121 :021,006,020,038,021,176,027
 5127 :211,006,020,038,021,176,223
 5133 :205,101,020,133,020,138,118
 5139 :101,021,176,196,133,021,155
 5145 :006,020,038,021,176,188,218
 5151 :152,101,020,133,020,165,110
 5157 :021,105,000,176,177,133,137
 5163 :021,076,225,166,165,020,204
 5169 :133,028,165,021,201,250,079
 5175 :144,007,169,220,162,172,161
 5181 :076,219,166,133,029,169,085
 5187 :000,133,026,169,208,133,224
 5193 :027,165,071,041,016,240,121
 5199 :029,032,021,171,032,083,191
 5205 :171,032,109,171,208,048,056
 5211 :169,000,141,183,173,032,021
 5217 :014,170,169,214,162,173,231
 5223 :032,065,170,076,135,167,236
 5229 :036,072,080,026,032,021,120
 5235 :171,032,083,171,032,109,201
 5241 :171,208,015,169,058,141,115
 5247 :183,173,032,014,170,169,100
 5253 :253,162,173,032,065,170,220
 5259 :032,014,170,165,065,032,105
 5265 :091,170,141,126,173,142,220
 5271 :125,173,165,066,032,091,035
 5277 :170,141,136,173,142,135,030
 5283 :173,165,070,041,002,240,086
 5289 :004,169,255,208,002,169,208
 5295 :050,141,143,173,032,083,029
 5301 :171,169,114,162,174,032,235
 5307 :155,165,032,249,169,208,141

5313 :007,169,118,162,173,076,130
 5319 :201,167,169,138,162,173,185
 5325 :032,065,170,032,108,170,014
 5331 :036,072,112,038,032,224,213
 5337 :169,160,039,032,204,169,222
 5343 :201,096,240,015,201,160,112
 5349 :240,004,201,224,208,011,093
 5355 :032,213,169,197,066,208,096
 5361 :004,169,032,145,022,136,237
 5367 :016,227,198,038,208,221,131
 5373 :032,224,169,169,016,133,228
 5379 :008,160,000,032,204,169,064
 5385 :133,009,076,014,168,169,066
 5391 :146,133,009,162,000,134,087
 5397 :084,202,134,036,032,014,011
 5403 :170,032,143,170,160,039,229
 5409 :032,204,169,201,032,208,111
 5415 :018,136,016,246,169,017,129
 5421 :032,046,170,032,154,170,137
 5427 :198,038,208,232,076,213,248
 5433 :168,132,037,230,036,164,056
 5439 :036,032,204,169,170,048,210
 5445 :014,036,009,048,021,169,110
 5451 :146,133,009,032,046,170,099
 5457 :076,091,168,036,009,016,221
 5463 :007,169,018,133,009,032,199
 5469 :046,170,138,036,072,080,123
 5475 :002,041,063,201,032,240,166
 5481 :018,164,036,032,213,169,225
 5487 :197,008,240,009,133,008,194
 5493 :168,185,112,175,032,046,067
 5499 :170,138,041,127,201,032,064
 5505 :144,008,201,064,144,010,188
 5511 :201,096,144,003,024,105,196
 5517 :032,024,105,032,032,046,156
 5523 :170,165,036,197,037,240,224
 5529 :022,165,084,201,071,144,072
 5535 :156,032,170,170,032,014,221
 5541 :170,032,143,170,169,000,081
 5547 :133,084,076,056,168,032,208
 5553 :154,170,198,038,240,028,237
 5559 :165,036,201,039,208,006,070
 5565 :032,170,170,076,014,168,051
 5571 :169,034,032,046,170,169,047
 5577 :000,032,046,170,032,108,077
 5583 :170,076,010,168,032,170,065
 5589 :170,032,108,170,165,032,122
 5595 :133,026,165,033,133,027,224
 5601 :169,034,162,173,032,155,182
 5607 :165,032,249,169,208,070,100
 5613 :165,039,133,020,165,040,031
 5619 :133,021,032,014,170,032,133
 5625 :108,170,169,150,162,173,157
 5631 :032,065,170,165,071,041,031
 5637 :016,240,021,032,091,171,064
 5643 :208,038,032,014,170,032,249
 5649 :108,170,169,233,162,173,008
 5655 :032,065,170,076,047,169,070
 5661 :036,072,080,018,032,091,102

5667 :171,208,013,032,014,170,131
 5673 :032,108,170,169,194,162,108
 5679 :173,032,065,170,165,063,203
 5685 :240,031,169,054,162,173,114
 5691 :032,155,165,166,028,165,002
 5697 :029,032,065,172,169,083,103
 5703 :162,173,032,155,165,166,156
 5709 :039,165,040,032,065,172,078
 5715 :076,180,166,165,039,208,149
 5721 :002,198,040,198,039,165,219
 5727 :040,201,250,144,006,032,000
 5733 :181,170,076,053,167,165,145
 5739 :026,056,233,000,133,016,059
 5745 :165,027,233,208,133,017,128
 5751 :165,045,133,014,024,101,089
 5757 :016,133,016,168,165,046,157
 5763 :133,015,101,017,133,017,035
 5769 :197,056,144,026,208,004,004
 5775 :196,055,144,020,169,087,046
 5781 :162,173,032,155,165,169,237
 5787 :021,162,175,032,155,165,097
 5793 :032,243,169,076,208,166,031
 5799 :162,005,149,045,202,148,110
 5805 :045,202,016,248,032,115,063
 5811 :165,165,026,133,014,165,079
 5817 :027,133,015,169,000,133,150
 5823 :095,169,208,133,096,032,156
 5829 :115,165,032,124,198,032,095
 5835 :181,170,076,150,169,032,213
 5841 :129,170,177,022,032,135,106
 5847 :170,096,032,129,170,177,221
 5853 :024,032,135,170,041,015,126
 5859 :096,169,025,133,038,169,089
 5865 :000,133,022,133,024,169,202
 5871 :248,133,023,169,252,133,173
 5877 :025,096,032,228,255,240,097
 5883 :251,096,032,228,255,240,073
 5889 :251,201,089,240,011,201,226
 5895 :013,240,007,008,169,078,010
 5901 :032,210,255,040,096,169,047
 5907 :255,032,046,170,032,046,088
 5913 :170,165,020,032,046,170,116
 5919 :165,021,032,046,170,032,241
 5925 :133,198,144,002,230,064,040
 5931 :230,020,208,002,230,021,242
 5937 :096,032,049,172,160,000,046
 5943 :145,026,032,057,172,230,205
 5949 :026,208,002,230,027,230,016
 5955 :084,096,133,016,134,017,035
 5961 :160,000,177,016,201,255,114
 5967 :240,003,032,046,170,230,032
 5973 :016,208,002,230,017,201,247
 5979 :000,208,235,096,162,255,023
 5985 :041,015,201,010,144,006,002
 5991 :162,049,024,105,038,096,065
 5997 :105,048,096,165,026,133,170
 6003 :032,165,027,133,033,165,158
 6009 :020,133,039,165,021,133,120
 6015 :040,165,064,133,063,096,176

6021 :120,169,037,133,001,096,177
 6027 :072,169,038,133,001,104,144
 6033 :088,096,169,153,032,046,217
 6039 :170,169,034,032,046,170,004
 6045 :096,169,040,024,101,022,097
 6051 :133,022,133,024,144,004,111
 6057 :230,023,230,025,096,169,174
 6063 :147,162,173,032,065,170,156
 6069 :032,108,170,096,169,153,141
 6075 :162,172,032,155,165,165,014
 6081 :040,166,039,032,065,172,195
 6087 :169,013,032,210,255,096,206
 6093 :169,000,240,002,169,015,032
 6099 :141,033,208,141,032,208,206
 6105 :169,040,141,024,208,169,200
 6111 :200,141,022,208,173,017,216
 6117 :208,041,016,009,011,141,143
 6123 :017,208,169,000,141,021,023
 6129 :208,096,133,015,132,017,074
 6135 :160,000,132,014,132,016,189
 6141 :177,014,145,016,136,208,181
 6147 :249,230,015,230,017,202,178
 6153 :208,242,096,056,032,240,115
 6159 :255,192,040,144,004,152,034
 6165 :233,040,168,096,032,083,161
 6171 :171,169,085,162,174,032,052
 6177 :155,165,032,249,169,208,243
 6183 :046,032,014,170,032,108,185
 6189 :170,165,067,032,091,170,228
 6195 :141,172,173,142,171,173,255
 6201 :165,068,032,091,170,141,212
 6207 :182,173,142,181,173,165,055
 6213 :069,032,091,170,141,192,252
 6219 :173,142,191,173,169,164,063
 6225 :162,173,032,065,170,096,011
 6231 :169,016,162,174,032,155,027
 6237 :165,096,169,145,162,174,236
 6243 :032,155,165,169,038,162,052
 6249 :174,032,155,165,032,249,144
 6255 :169,096,169,038,162,174,151
 6261 :032,155,165,032,249,169,151
 6267 :096,162,034,181,008,157,249
 6273 :144,175,202,016,248,162,052
 6279 :034,181,057,157,179,175,150
 6285 :202,016,248,162,214,160,119
 6291 :175,056,032,141,255,120,158
 6297 :032,138,255,169,239,141,103
 6303 :024,003,169,198,141,025,207
 6309 :003,169,154,141,020,003,143
 6315 :169,198,141,021,003,088,023
 6321 :169,000,141,039,208,133,099
 6327 :013,133,018,133,012,141,121
 6333 :023,208,173,001,003,201,030
 6339 :198,240,019,141,194,198,161
 6345 :173,000,003,141,193,198,141
 6351 :169,188,141,000,003,169,109
 6357 :198,141,001,003,169,004,217
 6363 :160,200,162,004,032,239,248
 6369 :170,173,002,221,009,003,035

6375 :141,002,221,169,000,141,137
 6381 :000,221,173,024,208,041,136
 6387 :002,009,040,141,024,208,155
 6393 :169,200,141,136,002,169,042
 6399 :028,141,248,203,160,004,015
 6405 :169,029,153,249,203,136,176
 6411 :208,250,169,030,141,254,039
 6417 :203,169,031,141,255,203,251
 6423 :160,007,185,128,175,153,063
 6429 :004,208,136,016,247,169,041
 6435 :060,141,016,208,169,028,145
 6441 :141,029,208,141,027,208,027
 6447 :169,019,032,210,255,096,060
 6453 :120,072,169,036,133,001,072
 6459 :104,096,072,169,038,133,159
 6465 :001,104,088,096,134,086,062
 6471 :133,087,162,003,169,000,113
 6477 :149,088,202,016,251,160,175
 6483 :015,006,086,038,087,120,179
 6489 :248,165,088,101,088,133,144
 6495 :088,165,089,101,089,133,248
 6501 :089,165,090,101,090,133,001
 6507 :090,216,088,136,016,227,112
 6513 :162,002,181,088,072,074,180
 6519 :074,074,074,032,138,172,171
 6525 :104,041,015,032,138,172,115
 6531 :202,016,237,165,091,208,026
 6537 :003,032,142,172,096,197,011
 6543 :091,240,251,009,048,133,147
 6549 :091,076,210,255,032,157,202
 6555 :157,000,147,031,002,010,246
 6561 :013,076,065,083,084,032,002
 6567 :076,073,078,069,032,078,061
 6573 :085,077,066,069,082,032,072
 6579 :085,083,069,068,058,032,062
 6585 :000,147,018,028,042,042,206
 6591 :042,032,073,078,086,065,055
 6597 :076,073,068,032,069,078,081
 6603 :084,082,089,032,042,042,062
 6609 :042,146,032,084,082,089,172
 6615 :032,065,071,065,073,078,087
 6621 :046,013,000,013,028,076,141
 6627 :073,078,069,032,078,085,130
 6633 :077,066,069,082,032,069,116
 6639 :088,067,069,069,068,083,171
 6645 :032,054,051,057,057,057,041
 6651 :032,076,073,077,073,084,154
 6657 :013,000,013,013,151,069,004
 6663 :078,084,069,082,032,066,162
 6669 :069,071,073,078,078,073,199
 6675 :078,071,032,076,073,078,171
 6681 :069,032,078,085,077,066,176
 6687 :069,082,032,079,082,013,132
 6693 :000,013,013,065,068,068,008
 6699 :032,087,065,073,084,032,160
 6705 :076,079,079,080,063,032,202
 6711 :089,157,000,147,028,002,222
 6717 :010,013,069,088,073,083,141
 6723 :084,073,078,071,032,076,225

6729 :073,078,069,083,032,066,218
 6735 :069,084,087,069,069,078,023
 6741 :032,000,032,038,032,000,219
 6747 :147,002,010,013,028,018,053
 6753 :042,042,042,032,079,085,163
 6759 :084,032,079,070,032,077,221
 6765 :069,077,079,082,089,032,025
 6771 :042,042,042,146,013,013,157
 6777 :000,151,053,051,050,056,226
 6783 :048,044,048,048,058,151,012
 6789 :053,051,050,056,049,044,180
 6795 :048,048,058,153,199,040,173
 6801 :049,052,050,041,034,147,006
 6807 :034,059,000,151,049,057,245
 6813 :056,044,048,058,146,049,046
 6819 :057,056,044,049,000,151,008
 6825 :053,051,050,056,050,044,217
 6831 :048,048,058,151,053,051,072
 6837 :050,056,051,044,048,048,222
 6843 :058,151,053,051,050,056,094
 6849 :052,044,048,048,000,151,024
 6855 :053,051,050,054,053,044,248
 6861 :194,040,053,051,050,054,135
 6867 :053,041,175,049,057,049,123
 6873 :000,151,053,051,050,055,065
 6879 :048,044,194,040,053,051,141
 6885 :050,055,048,041,176,049,136
 6891 :054,000,151,053,051,050,082
 6897 :055,048,044,194,040,053,163
 6903 :051,050,055,048,041,175,155
 6909 :050,051,057,000,151,053,103
 6915 :051,050,054,053,044,194,193
 6921 :040,053,051,050,054,053,054
 6927 :041,176,054,052,000,013,095
 6933 :013,065,068,068,032,076,087
 6939 :073,078,069,032,084,079,186
 6945 :032,083,069,084,032,085,162
 6951 :080,013,000,069,088,084,117
 6957 :069,078,068,069,068,032,173
 6963 :066,065,067,075,071,082,221
 6969 :079,085,078,068,032,077,220
 6975 :079,068,069,032,079,082,216
 6981 :013,077,085,076,084,073,221
 6987 :067,079,076,082,032,077,232
 6993 :079,068,069,063,032,089,225
 6999 :157,000,066,065,067,075,005
 7005 :071,082,079,085,078,068,044
 7011 :032,082,069,071,073,083,253
 7017 :084,069,082,083,032,049,248
 7023 :045,051,063,032,089,157,036
 7029 :000,066,065,067,075,071,205
 7035 :082,079,085,078,068,032,035
 7041 :038,032,066,079,082,068,238
 7047 :069,082,032,067,079,076,028
 7053 :079,082,083,063,032,089,057
 7059 :157,000,013,013,065,068,207
 7065 :068,032,076,073,078,069,037
 7071 :032,084,079,032,084,085,043
 7077 :082,078,032,079,070,070,064

7083 :013,000,147,013,028,018,134
 7089 :002,005,032,080,082,069,191
 7095 :083,083,032,075,069,089,102
 7101 :083,032,049,045,051,032,225
 7107 :084,079,032,067,089,067,101
 7113 :076,069,032,067,079,076,088
 7119 :079,082,083,002,005,032,234
 7125 :146,002,003,013,000,013,134
 7131 :151,002,029,032,111,002,034
 7137 :004,183,112,013,002,005,032
 7143 :032,066,065,067,075,071,095
 7149 :082,079,085,078,068,032,149
 7155 :082,069,071,073,083,084,193
 7161 :069,082,032,032,048,032,032
 7167 :032,165,002,004,032,167,145
 7173 :013,002,029,032,108,002,191
 7179 :004,175,186,013,013,000,146
 7185 :013,013,013,018,031,002,107
 7191 :006,032,080,082,069,083,119
 7197 :083,032,039,083,080,065,155
 7203 :067,069,039,032,070,079,135
 7209 :082,032,087,079,082,075,222
 7215 :045,083,067,082,069,069,206
 7221 :078,002,005,032,146,000,060
 7227 :137,138,139,140,133,134,112
 7233 :135,136,002,020,013,015,130
 7239 :008,016,024,026,011,003,159
 7245 :021,161,053,161,059,162,182
 7251 :043,161,205,161,176,161,222
 7257 :065,161,161,161,080,161,110
 7263 :003,161,024,162,143,164,240
 7269 :021,164,031,166,162,166,043
 7275 :005,161,095,161,102,161,024
 7281 :125,161,148,144,005,028,212
 7287 :159,156,030,031,158,129,014
 7293 :149,150,151,152,153,154,010
 7299 :155,000,099,000,139,000,012
 7305 :179,080,242,032,142,198,242
 7311 :076,029,164,032,147,198,021
 7317 :076,105,254,032,147,198,193
 7323 :032,051,165,076,142,198,051
 7329 :032,147,198,032,019,166,243
 7335 :076,142,198,169,038,133,155
 7341 :001,096,072,169,039,133,171
 7347 :001,104,096,165,018,240,035
 7353 :027,198,019,208,023,169,061
 7359 :020,133,019,173,039,208,015
 7365 :073,001,141,039,208,173,064
 7371 :045,208,073,001,141,045,204
 7377 :208,141,046,208,076,049,169
 7383 :234,224,011,240,003,076,235
 7389 :139,227,165,122,141,219,210
 7395 :198,165,123,141,220,198,248
 7401 :160,004,173,219,198,208,171
 7407 :003,206,220,198,206,219,011
 7413 :198,173,001,008,217,234,052
 7419 :198,208,222,136,016,234,241
 7425 :104,104,076,112,198,071,154
 7431 :069,078,073,069,064,170,018
 7437 :000,000,000,000,255,255,011

MLX

di C. Brannon

trad. e adatt. di M. Cristuib Grizzi

M LX é un programma che permette di inserire listati in linguaggio macchina esenti in modo assoluto da errori e senza la perdita di tempo del dover battere e controllare lunghe sequenze di istruzioni DATA.

Molti dei nostri listati di programmi in linguaggio macchina hanno il formato MLX (li riconoscete dal fatto che sono esclusivamente numerici) e richiedono quindi che MLX sia caricato in memoria ed eseguito prima della battitura del listato.

MLX vi chiederà l'indirizzo della locazione finale del programma da caricare.

Questi valori sono sempre indicati nell'articolo che accompagna il listato.

MLX vi segnala automaticamente gli errori di battitura MENTRE STATE DIGITANDO IL LISTATO e vi chiede di reinserire la linea errata. L'ultimo numero battuto di ogni linea rappresenta il checksum e viene visualizzato in reverse.

Sono inoltre disponibili altri comandi, quali:

SHIFT-N nuovo indirizzo: permette di cambiare l'indirizzo della linea che volete battere ed é utile nel caso si inseriscano i listati in più parti.

SHIFT-D display: lista i dati inseriti tra due indirizzi di inizio e fine.

SHIFT-L load: carica un file da nastro o SHIFT-S save: salva su nastro o disco un file in formato MLX.

Una volta battuto il listato, e salvatolo tramite MLX, si potrà caricare direttamente il programma con un'istruzione LOAD "nome del file",1,1 per il registratore, oppure LOAD "nome file",8,1 per l'unità a dischi, salvo diverse indicazioni date negli articoli corrispondenti ai vari programmi. Il programma sarà generalmente mandato in esecuzione con una SYS (indirizzo di partenza).

La versione originaria di MLX per C64 ha subito, dalla prima pubblicazione, diversi ritocchi e miglioramenti, fino alla versione 2.0 qui presentata.

E' stato in primo luogo modificato il colo-

re di fondo dello schermo e del bordo, in modo da risultare meno stancante alla vista; sono state ulteriormente perfezionate le routine in linguaggio macchina di salvataggio e caricamento dei programmi; é cambiata la forma del cursore e, cosa più importante, é stato aggiunto un tastierino numerico per gli utenti che si trovano più a loro agio con una diversa disposizione dei tasti.

Oltre a poter usare i soliti tasti numerici, sono stati ridefiniti alcuni tasti alfabetici, in modo da generare ugualmente dei numeri. I tasti ridefiniti sono i seguenti:

U I O		7 8 9
H J K L	diventano	0 4 5 6
M , .		1 2 3

Le persone abituate a usare tali tastierini accresceranno senz'altro, in brevissimo tempo, la velocità di battitura dei programmi in linguaggio macchina.

MLX

```
100 PRINT"{CLR}{CYN}";CHR$(142);CHR$(8);:
    POKE53280,0:POKE53281,0      :rem 71
101 POKE788,52:REM DISABILITA RUN/STOP
                                :rem 89
110 PRINT"{RVS}{ 40 SPAZI}";    :rem 176
120 PRINT"{RVS}{ 15 SPAZI}{DES}{OFF}{[<*>]}
    £{RVS}{DES}{DES}{ 2 SPAZI}{[<*>]}{OFF}
    [<*>]£{RVS}£{RVS}{ 13 SPAZI}";
                                :rem 250
130 PRINT"{RVS}{ 15 SPAZI}{DES}{[<N>]}{[<H>]}
    {DES}{ 2 DES}{OFF}£{RVS}£[<*>]}{OFF}
    [<*>]{RVS}{ 13 SPAZI}";      :rem 220
140 PRINT"{RVS}{ 40 SPAZI}"     :rem 120
200 PRINT"{ 2 GIU'}{PUR}EDITOR DI CODICE
    MACCHINA VERSIONE 2.0{ 3 GIU'}"
                                :rem 126
210 PRINT"[<5>]{ 2 SU} INDIRIZZO DI PARTE
    NZA{ 2 SPAZI}";:INPUTS:F=1-F:C$=CHR$(
    31+119*F)                   :rem 85
220 IFS<256OR(S>40960ANDS<49152)ORS>53247
    THENGOSUB3000:GOTO210       :rem 235
225 PRINT:PRINT:PRINT          :rem 180
230 PRINT"[<5>]{ 2 SU} INDIRIZZO CONCLUSI
    VO{ 3 SPAZI}";:INPUTE:F=1-F:C$=CHR$(3
    1+119*F)                   :rem 92
240 IFE<256OR(E>40960ANDE<49152)ORE>53247
    THENGOSUB3000:GOTO230       :rem 183
250 IFE<STHENPRINTC$;"{RVS}INDIRIZZO CONC
```

```
LUSIVO<INDIRIZZO INIZIALE" :rem 200
255 IFE<STHENGOSUB1000:GOTO230 :rem 119
260 PRINT:PRINT:PRINT       :rem 179
300 PRINT"{CLR}";CHR$(14):AD=S:POKEV+21,0
                                :rem 225
310 A=1:PRINTRIGHT$("0000"+MID$(STR$(AD),
    2),5);":":FORJ=ATO6        :rem 227
315 FORJ=ATO6                 :rem 33
320 GOSUB570:IFN=-1THENJ=J+N:GOTO320
                                :rem 228
390 IFN=-211THEN710           :rem 62
400 IFN=-204THEN790           :rem 64
410 IFN=-206THENPRINT:INPUT"{GIU'} INSERI
    RE IL NUOVO INDIRIZZO";ZZ :rem 13
414 IFN=-206THENIFZZ<SORZZ>ETHENPRINT"
    {RVS}ESCE DAL CAMPO DI VALORI INDICAT
    O"                          :rem 105
415 IFN=-206THENIFZZ<SORZZ>ETHENGOSUB1000
    :GOTO410                   :rem 202
417 IFN=-206THENAD=ZZ:PRINT:GOTO310
                                :rem 238
420 IFN<>-196THEN480           :rem 133
430 PRINT:INPUT"LISTATO:DA";F:PRINT"
    { 9 SPAZI}A";:INPUTT      :rem 183
440 IFF<SORF>EORT<SORT>ETHENPRINT"MINIMO"
    ;S;" MASSIMO";E;"! [<5>]":GOTO430
                                :rem 147
450 FORI=FTOTSTEP6:PRINT:PRINTRIGHT$("000
```



```

0"+MID$(STR$(I),2),5);":": :rem 30
451 FORK=0TO5:N=PEEK(I+K):PRINTRIGHT$( "00
"+MID$(STR$(N),2),3);":": :rem 66
460 GETA$:IFA$>" "THENPRINT:PRINT:GOTO310
:rem 25
470 NEXTK:PRINTCHR$(20);:NEXTI:PRINT:PRIN
T:GOTO310 :rem 50
480 IFN<0THENPRINT:GOTO310 :rem 168
490 A(J)=N:NEXTJ :rem 199
500 CKSUM=AD-INT(AD/256)*256:FORI=1TO6:CK
SUM=(CKSUM+A(I))AND255:NEXT :rem 200
510 PRINTCHR$(18);:GOSUB570:PRINTCHR$(146
); :rem 94
511 IFN=-1THENA=6:GOTO315 :rem 254
515 PRINTCHR$(20):IFN=CKSUMTHEN530
:rem 122
520 PRINT:PRINT"{RED}LA LINEA E' STATA IN
SERITA IN MANIERA" :rem 157
525 PRINT"ERRATA. RIPETERE[<5>]":PRINT:GO
SUB1000:GOTO310 :rem 27
530 GOSUB2000 :rem 218
540 FORI=1TO6:POKEAD+I-1,A(I):NEXT:POKE54
272,0:POKE54273,0 :rem 227
550 AD=AD+6:IFAD<ETHEN310 :rem 212
560 GOTO710 :rem 108
570 N=0:Z=0 :rem 88
580 PRINT"[<E>]"; :rem 81
581 GETA$:IFA$=""THEN581 :rem 95
582 AV=- (A$="M")-2*(A$=",")-3*(A$=".")-4*
(A$="J")-5*(A$="K")-6*(A$="L")
:rem 41
583 AV=AV-7*(A$="U")-8*(A$="I")-9*(A$="O"
):IFA$="H"THENA$="0" :rem 134
584 IFAV>0THENA$=CHR$(48+AV) :rem 134
585 PRINTCHR$(20);:A=ASC(A$):IFA=13ORA=44
ORA=32THEN670 :rem 229
590 IFA>128THENN=-A:RETURN :rem 137
600 IFA<>20THEN630 :rem 10
610 GOSUB690:IFI=1ANDT=44THENN=-1:PRINT"
{SIN} {SIN}";:GOTO690 :rem 172
620 GOTO570 :rem 109
630 IFA<48ORA>57THEN580 :rem 105
640 PRINTA$;:N=N*10+A-48 :rem 106
650 IFN>255THENA=20:GOSUB1000:GOTO600
:rem 229
660 Z=Z+1:IFZ<3THEN580 :rem 71
670 IFZ=0THENGOSUB1000:GOTO570 :rem 114
680 PRINT",";:RETURN :rem 240
690 S%=PEEK(209)+256*PEEK(210)+PEEK(211)
:rem 149
691 FORI=1TO3:T=PEEK(S%-I) :rem 67
695 IFT<>44ANDT<>58THENPOKES%-I,32:NEXT
:rem 205
700 PRINTLEFT$("{ 3 SIN}",I-1);:RETURN
:rem 7
710 PRINT"{CLR}{RVS}*** SAVE ***
{ 3 GIU'}" :rem 236
715 PRINT"{ 2 GIU'}PREMERE {RVS}RETURN
{OFF} PER USCIRE DAL SAVE{GIU'}"
:rem 103
720 F$="":INPUT"{GIU'}NOME DEL FILE";F$:I
FF$=""THENPRINT:PRINT:GOTO310
:rem 42
730 PRINT:PRINT"{ 2 GIU'}{RVS}N{OFF}ASTRO
O {RVS}D{OFF}ISCO: (N/D)" :rem 128
740 GETA$:IFA$<>"N"ANDA$<>"D"THEN740
:rem 30
750 DV=1-7*(A$="D"):IFDV=8THENF$="0:"+F$:
OPEN15,8,15,"S"+F$:CLOSE15 :rem 212
760 T$=F$:ZK=PEEK(53)+256*PEEK(54)-LEN(T$
):POKE782,ZK/256 :rem 3

```

```

762 POKE781,ZK-PEEK(782)*256:POKE780,LEN(
T$):SYS65469 :rem 109
763 POKE780,1:POKE781,DV:POKE782,1:SYS654
66 :rem 69
765 K=S:POKE254,K/256:POKE253,K-PEEK(254)
*256:POKE780,253 :rem 17
766 K=E+1:POKE782,K/256:POKE781,K-PEEK(78
2)*256:SYS65496 :rem 235
770 IF(PEEK(783)AND1)OR(191ANDST)THEN780
:rem 111
775 PRINT"{GIU'}FATTO.{GIU'}":GOTO310
:rem 201
780 PRINT"{GIU'}ERRORE NEL SAVE-RIPROVA!"
:IFDV=1THEN720 :rem 104
781 OPEN15,8,15:INPUT#15,E1$,E2$:PRINT E1$
;E2$:CLOSE15:GOTO720 :rem 103
790 PRINT"{CLR}{RVS}*** LOAD ***
{ 2 GIU'}" :rem 212
795 PRINT"{ 2 GIU'}PREMERE {RVS}RETURN
{OFF} PER USCIRE DAL LOAD{GIU'}"
:rem 96
800 F$="":INPUT"{ 2 GIU'}NOME DEL FILE";F
$:IFF$=""THENPRINT:GOTO310 :rem 115
810 PRINT:PRINT"{ 2 GIU'}{RVS}N{OFF}ASTRO
O {RVS}D{OFF}ISCO: (N/D)" :rem 127
820 GETA$:IFA$<>"N"ANDA$<>"D"THEN820
:rem 28
830 DV=1-7*(A$="D"):IFDV=8THENF$="0:"+F$
:rem 157
840 T$=F$:ZK=PEEK(53)+256*PEEK(54)-LEN(T$
):POKE782,ZK/256 :rem 2
841 POKE781,ZK-PEEK(782)*256:POKE780,LEN(
T$):SYS65469 :rem 107
845 POKE780,1:POKE781,DV:POKE782,1:SYS654
66 :rem 70
850 POKE780,0:SYS65493 :rem 11
860 IF(PEEK(783)AND1)OR(191ANDST)THEN870
:rem 111
865 PRINT"{GIU'}FATTO.":GOTO310 :rem 184
870 PRINT"{GIU'}ERRORE NEL LOAD-RIPETI!
{GIU'}":IFDV=1THEN800 :rem 19
880 OPEN15,8,15:INPUT#15,E1$,E2$:PRINT E1$
;E2$:CLOSE15:GOTO800 :rem 102
1000 REM CICALINO :rem 231
1001 POKE54296,15:POKE54277,45:POKE54278,
165 :rem 207
1002 POKE54276,33:POKE54273,6:POKE54272,5
:rem 42
1003 FORT=1TO200:NEXT:POKE54276,32:POKE54
273,0:POKE54272,0:RETURN :rem 202
2000 REM CAMPANELLO :rem 130
2001 POKE54296,15:POKE54277,0:POKE54278,2
47 :rem 152
2002 POKE54276,17:POKE54273,40:POKE54272,
0 :rem 86
2003 FORT=1TO100:NEXT:POKE54276,16:RETURN
:rem 57
3000 PRINTC$;"{RVS} NON IN PAGINA ZERO O
SU{DES}ROM ":GOTO1000 :rem 240

```


WORK

Help!

di J. Siddiqui
trad. e adatt. di E. Stefani

Help!

Questa interessante routine in linguaggio macchina vi fornisce un comodo strumento per inserire fino a otto schermate di aiuto nei vostri programmi, siano essi in BASIC oppure in linguaggio macchina. E' richiesto l'uso del disk drive.

Molti dei potenti programmi di utilità per il C64 e C128 possiedono un gran numero di comandi e opzioni diverse; in molti casi risulta difficile, se non impossibile, tenere a mente la sintassi di tutti i comandi disponibili, e dover ricorrere al manuale oppure all'articolo sulla rivista relativo al programma risulta spesso una procedura alquanto scomoda.

"Help!" vi offre una soluzione molto comoda: seguendo una semplice procedura, potete ora creare fino a otto schermate contenenti le descrizioni dei comandi (o ciò che preferite...) da inserire nei vostri programmi.

Dando il RUN ad "Help!" ci sarà una breve attesa, necessaria ad inserire in memoria il codice macchina, poi ricomparirà il cursore con la familiare scritta READY in uno schermo vuoto.

Premete CLR per pulire lo schermo, e digitate le informazioni che desiderate inserire nella prima delle schermate di aiuto (*non premete mai RETURN per andare a capo, altrimenti verrà generato un SYNTAX ERROR: spostatevi invece liberamente con l'uso dei tasti cursore!*). Evitate inoltre di premere il tasto cursore verso il basso quando vi trovate sul-

l'ultima riga, altrimenti lo schermo eseguirà uno scroll verso l'alto, facendovi perdere la riga più alta.

Quando avete terminato di scrivere il testo sullo schermo premete contemporaneamente i tasti CTRL ed F. Se usate un C128, premete il tasto ENTER sul tastierino numerico.

A questo punto "Help!" immagazzinerà in memoria la videata appena creata: la versione per C64 esegue un lampeggio del bordo dello schermo per segnalarlo, mentre la versione per C128 lo modifica. Dopo aver salvato in memoria la prima videata potete premere nuovamente CLR

per pulire lo schermo e iniziare a scrivere la seconda con lo stesso procedimento (e così sarà per tutte e otto le schermate).

Una volta premuto CTRL+F (o ENTER sul tastierino per il C128) in corrispondenza dell'ottava schermata viene automaticamente creato su disco un programma dal nome "HELPER.EXE", mentre le schermate vengono salvate in un file su disco dal nome "HS".

Se desiderate avere un numero di schermate inferiore a otto, nella versione per C64 premete contemporaneamente i tasti CTRL e D dopo aver completato il vo-



stro ultimo schermo, mentre nella versione per C128 premete contemporaneamente i tasti SHIFT ed ENTER. E' un'ottima idea quella di salvare le schermate di aiuto sullo stesso disco che contiene il programma cui le schermate si riferiscono.

Help in linea

Vediamo ora come utilizzare le schermate di aiuto appena generate in congiunzione al programma cui le schermate si riferiscono: per prima cosa, se avete il C64, digitate LOAD"HELPER.EXE", 8,1 mentre se usate il C128, digitate BLOAD"HELPER.EXE". Digitate quindi SYS5632.

"Help!" funziona con tutti i programmi in BASIC e con la maggior parte di quelli in linguaggio macchina; dopo avergli dato la SYS esso provvede a caricare il file HS contenente le schermate. Digitate quindi un bel NEW, e caricate e attivate il programma cui le schermate si riferiscono.

D'ora in poi, in qualunque momento desideriate richiamare le schermate di aiuto sarà sufficiente premere contemporaneamente i tasti CTRL e H (per il C128 premere il tasto HELP). Verrà visualizzata la prima videata di aiuto, e per passare alle videate successive premete per il C64 i tasti CTRL e cursore a destra (CTRL e cursore in alto riporta alla pagina precedente), mentre per il C128 pre-

mete i tasti con freccia verso il basso o verso l'alto per muovervi tra le videate.

Per ritornare al programma premete contemporaneamente i tasti CTRL e DEL per il C64, mentre per il C128 premete nuovamente il tasto HELP.

Se richiamerete ancora le videate di aiuto, verrà automaticamente visualizzata la videata su cui ci si trovava la volta precedente.

Si noti che la pressione di RUN/STOP e RESTORE oppure l'esecuzione di certi programmi in linguaggio macchina non compatibili disabilitano la visualizzazione delle videate di aiuto. Nella maggior parte dei casi questa può essere riabilitata digitando SYS679 per il C64, oppure SYS5728 per il C128.

Help! versione per C64

10 FORS=679TO766:READA:POKES,A:NEXT:FORS=	220 DATA 032,208,165,197,201,064,208	
820TO1016:READA:POKES,A:NEXT :rem	:rem	233
20 FORS=49152TO49397:READA:POKES,A:NEXT:P	230 DATA 247,238,032,208,173,141,002	
RINT"{CLR}{WHT}PROGRAMMA ATTIVATO"	:rem	227
:rem	240 DATA 201,004,208,246,165,197,201	
30 SYS49152:NEW :rem	:rem	229
40 DATA 173,245,003,208,018,173,020	250 DATA 002,240,217,201,007,240,223	
:rem	:rem	210
50 DATA 003,141,212,002,173,021,003	260 DATA 201,000,208,232,169,063,141	
:rem	:rem	221
60 DATA 141,213,002,238,245,003,032	270 DATA 002,221,169,199,141,000,221	
:rem	:rem	224
70 DATA 223,003,160,002,169,214,032	280 DATA 169,021,141,024,208,104,141	
:rem	:rem	224
80 DATA 160,003,088,169,131,141,002	290 DATA 032,208,104,141,021,208,160	
:rem	:rem	219
90 DATA 003,169,164,141,003,003,076	300 DATA 002,169,214,032,160,003,169	
:rem	:rem	221
100 DATA 116,164,076,000,000,173,141	310 DATA 071,141,024,003,076,211,002	
:rem	:rem	208
110 DATA 002,201,004,208,246,165,197	320 DATA 120,141,020,003,140,021,003	
:rem	:rem	192
120 DATA 201,029,208,240,160,234,169	330 DATA 096,173,246,003,176,010,105	
:rem	:rem	230
130 DATA 049,032,160,003,088,169,193	340 DATA 016,208,002,169,128,141,246	
:rem	:rem	233
140 DATA 141,024,003,173,002,221,009	350 DATA 003,096,233,016,201,128,176	
:rem	:rem	231
150 DATA 003,141,002,221,173,021,208	360 DATA 246,169,240,144,242,173,024	
:rem	:rem	239
160 DATA 072,076,052,003,169,000,141	370 DATA 208,041,015,013,246,003,141	
:rem	:rem	218
170 DATA 021,208,169,197,141,000,221	380 DATA 024,208,096,162,000,157,000	
:rem	:rem	226
180 DATA 173,032,208,072,173,134,002	390 DATA 216,157,255,216,157,254,217	
:rem	:rem	251
190 DATA 032,205,003,076,079,003,024	400 DATA 157,253,218,232,208,241,096	
:rem	:rem	239
200 DATA 032,168,003,032,193,003,076	410 DATA 169,001,162,008,168,032,186	
:rem	:rem	236
210 DATA 089,003,056,076,076,003,238	420 DATA 255,169,002,162,247,160,003	
:rem	:rem	230
	430 DATA 032,189,255,169,000,076,213	
	:rem	239


```

440 DATA 255,000,128,072,083,169,017
      :rem 237
450 DATA 160,192,032,201,192,088,169
      :rem 242
460 DATA 193,141,024,003,096,076,049
      :rem 241
470 DATA 234,173,141,002,201,004,208
      :rem 218
480 DATA 246,165,197,201,018,240,071
      :rem 241
490 DATA 201,021,208,236,032,209,192
      :rem 230
500 DATA 032,197,192,088,169,000,133
      :rem 240
510 DATA 095,169,004,133,096,169,232
      :rem 247
520 DATA 133,090,169,007,133,091,173
      :rem 236
530 DATA 244,192,133,088,173,245,192
      :rem 249
540 DATA 133,089,032,191,163,238,032
      :rem 239
550 DATA 208,162,000,032,179,238,232
      :rem 233
560 DATA 208,250,206,032,208,173,231
      :rem 230
570 DATA 192,201,008,240,010,169,017
      :rem 228
580 DATA 160,192,032,201,192,076,014
      :rem 232
590 DATA 192,032,197,192,088,169,167
      :rem 12
600 DATA 141,002,003,133,253,169,002
      :rem 214
610 DATA 141,003,003,133,254,032,222
      :rem 210
620 DATA 192,169,010,162,232,160,192
      :rem 234
630 DATA 032,189,255,162,250,160,003
      :rem 232
640 DATA 169,253,032,216,255,032,222
      :rem 235
650 DATA 192,169,002,162,242,160,192
      :rem 239
660 DATA 032,189,255,169,000,133,253
      :rem 242
670 DATA 169,160,133,254,165,001,041
      :rem 234
680 DATA 254,133,001,169,253,162,255
      :rem 242
690 DATA 160,191,032,216,255,169,071
      :rem 245
700 DATA 141,024,003,169,131,141,002
      :rem 213
710 DATA 003,169,164,141,003,003,165
      :rem 224
720 DATA 001,009,001,133,001,096,169
      :rem 221
730 DATA 049,160,234,120,141,020,003
      :rem 215
740 DATA 140,021,003,096,173,245,192
      :rem 233
750 DATA 024,105,004,141,245,192,238
      :rem 232

```

```

760 DATA 231,192,096,169,001,162,008
      :rem 242
770 DATA 160,001,076,186,255,000,072
      :rem 233
780 DATA 069,076,080,069,082,046,069
      :rem 7
790 DATA 088,069,072,083,232,159:rem 63

```

Help! versione per C128

```

10 PRINT"{CLR}{2 SPAZI}MEMORIZZAZIONE L/{
   M"      :rem 208
20 FOR I=3072 TO 3249:READ A:POKE I,A:NEX
   T      :rem 63
30 FOR I=5632 TO 6045:READ A:POKE I,A:NEX
   T      :rem 65
40 BSAVE "HELPER.EXE",P5632 TO P6046
      :rem 248
50 PRINT "PROGRAMMA ATTIVATO":SYS3072
      :rem 77
60 END      :rem 61
3072 DATA 173,060,003,141,033,012,173
      :rem 12
3079 DATA 061,003,141,034,012,169,028
      :rem 25
3086 DATA 141,060,003,169,012,141,061
      :rem 18
3093 DATA 003,169,192,141,076,012,096
      :rem 37
3100 DATA 192,076,240,003,076,255,255
      :rem 29
3107 DATA 224,001,240,079,224,000,208
      :rem 14
3114 DATA 245,173,041,010,164,236,145
      :rem 23
3121 DATA 224,169,004,141,073,012,162
      :rem 17
3128 DATA 000,120,173,006,213,009,003
      :rem 6
3135 DATA 141,006,213,169,127,141,000
      :rem 16
3142 DATA 255,189,000,004,157,000,192
      :rem 23
3149 DATA 232,208,247,238,073,012,238
      :rem 40
3156 DATA 076,012,173,073,012,201,008
      :rem 20
3163 DATA 208,234,169,000,141,000,255
      :rem 20
3170 DATA 173,006,213,041,252,141,006
      :rem 14
3177 DATA 213,088,238,032,208,173,076
      :rem 46
3184 DATA 012,201,224,240,001,096,169
      :rem 22
3191 DATA 001,162,000,032,104,255,169
      :rem 16
3198 DATA 000,162,008,168,032,186,255
      :rem 39
3205 DATA 169,004,162,174,160,012,032
      :rem 20
3212 DATA 189,255,169,000,133,253,169
      :rem 41

```


3219 DATA 192,133,254,169,253,162,255	:rem	46	3443 DATA 072,169,000,141,033,208,173	:rem	26
3226 DATA 160,223,032,216,255,120,173	:rem	21	3450 DATA 002,221,009,003,141,002,221	:rem	254
3233 DATA 033,012,141,060,003,173,034	:rem	7	3457 DATA 173,000,221,072,041,252,009	:rem	21
3240 DATA 012,141,061,003,088,096,048	:rem	25	3464 DATA 000,141,000,221,173,006,213	:rem	5
3247 DATA 058,072,083,169,000,141,000	:rem	25	3471 DATA 072,041,063,009,067,141,006	:rem	26
3254 DATA 255,173,006,213,009,003,141	:rem	21	3478 DATA 213,032,108,023,088,165,212	:rem	34
3261 DATA 006,213,169,079,141,000,255	:rem	28	3485 DATA 201,088,208,250,165,212,201	:rem	30
3268 DATA 162,000,189,000,208,157,000	:rem	23	3492 DATA 064,240,024,201,083,240,007	:rem	21
3275 DATA 224,232,208,247,238,022,022	:rem	30	3499 DATA 201,084,208,242,169,240,044	:rem	42
3282 DATA 238,025,022,173,022,022,201	:rem	15	3506 DATA 169,016,024,109,044,010,041	:rem	21
3289 DATA 224,208,234,169,000,141,000	:rem	25	3513 DATA 127,141,044,010,016,220,165	:rem	14
3296 DATA 255,173,006,213,041,252,141	:rem	29	3520 DATA 212,201,088,208,250,120,169	:rem	24
3303 DATA 006,213,169,001,162,000,032	:rem	6	3527 DATA 022,141,007,213,173,006,213	:rem	17
3310 DATA 104,255,169,001,162,008,160	:rem	19	3534 DATA 009,003,141,006,213,169,112	:rem	19
3317 DATA 000,032,186,255,169,002,162	:rem	27	3541 DATA 141,000,255,160,000,198,123	:rem	16
3324 DATA 119,160,022,032,189,255,169	:rem	39	3548 DATA 198,125,177,124,145,122,200	:rem	39
3331 DATA 000,162,000,160,192,032,213	:rem	3	3555 DATA 208,249,165,123,201,216,208	:rem	37
3338 DATA 255,169,192,133,058,032,129	:rem	48	3562 DATA 239,169,000,141,000,255,141	:rem	24
3345 DATA 175,173,060,003,141,131,022	:rem	17	3569 DATA 007,213,104,141,006,213,104	:rem	18
3352 DATA 173,061,003,141,132,022,169	:rem	20	3576 DATA 141,000,221,104,141,033,208	:rem	13
3359 DATA 126,141,060,003,169,022,141	:rem	24	3583 DATA 104,141,021,208,173,044,010	:rem	17
3366 DATA 061,003,096,072,083,008,000	:rem	25	3590 DATA 141,121,022,104,141,044,010	:rem	5
3373 DATA 216,000,240,201,132,240,003	:rem	3	3597 DATA 104,133,217,104,133,216,104	:rem	26
3380 DATA 076,255,255,169,118,141,060	:rem	43	3604 DATA 141,025,003,104,141,024,003	:rem	3
3387 DATA 003,120,173,024,003,072,169	:rem	26	3611 DATA 076,108,022,169,112,141,000	:rem	17
3394 DATA 051,141,024,003,173,025,003	:rem	15	3618 DATA 255,169,022,141,007,213,160	:rem	30
3401 DATA 072,169,255,141,025,003,165	:rem	27	3625 DATA 000,177,122,145,124,169,013	:rem	27
3408 DATA 216,072,169,000,133,216,165	:rem	31	3632 DATA 145,122,200,208,245,230,123	:rem	17
3415 DATA 217,072,169,004,133,217,173	:rem	34	3639 DATA 230,125,165,123,201,220,208	:rem	23
3422 DATA 044,010,072,173,121,022,141	:rem	8	3646 DATA 235,169,000,141,000,255,141	:rem	23
3429 DATA 044,010,173,021,208,072,169	:rem	30	3653 DATA 007,213,173,006,213,041,252	:rem	21
3436 DATA 000,141,021,208,173,033,208	:rem	16	3660 DATA 141,006,213,096	:rem	206

WORK

Speed script 80 per C128

di T. Heimark

trad. e adatt. di S. Colombo

SpeedScript-80 per C128

Ora anche i possessori di C128 con monitor a 80 colonne possono utilizzare il nostro word processor SpeedScript: ecco le modifiche da apportare al programma per fare in modo che funzioni con 80 colonne. Ricordiamo che SpeedScript è stato pubblicato in versione 3.0 su SuperCommodore N. 9 del 1985.

SpeedScript già funziona egregiamente sul C128 in modo 64, ma, dato che il C128 supporta pienamente le 80 colonne, era fino a questo momento un peccato utilizzare il word processor soltanto sulle 40 colonne tipiche del modo 64. SpeedScript-80 si inserisce nello SpeedScript 3.0 originale per fornirvi un perfetto schermo a 80 colonne: esso va comunque eseguito in modo 64, dal momento che sfrutta ugualmente le caratteristiche del chip a 80 colonne (pochi sanno che tale chip è accessibile anche in modo 64...).

Naturalmente, SpeedScript-80 non funziona sui C64, dal momento che questi non possiedono il chip a 80 colonne.

Procedura di installazione

Si presume siate in possesso di SpeedScript 3.0 (la versione precedente non è adatta ad essere modificata con SpeedScript-80); spegnete e accendete il C128, ed entrate in monitor digitando il comando MONITOR. Inserite quindi il disco contenente SpeedScript 3.0 e digitate:

L "nome prog",8,02801

dove "nome prog" è il nome che avete assegnato a SpeedScript 3.0.

Ora caricate dalla cassetta allegata alla rivista (o dalla vostra cassetta se li avete digitati) i primi quattro programmi in fondo a questo articolo, rispettivamente con:

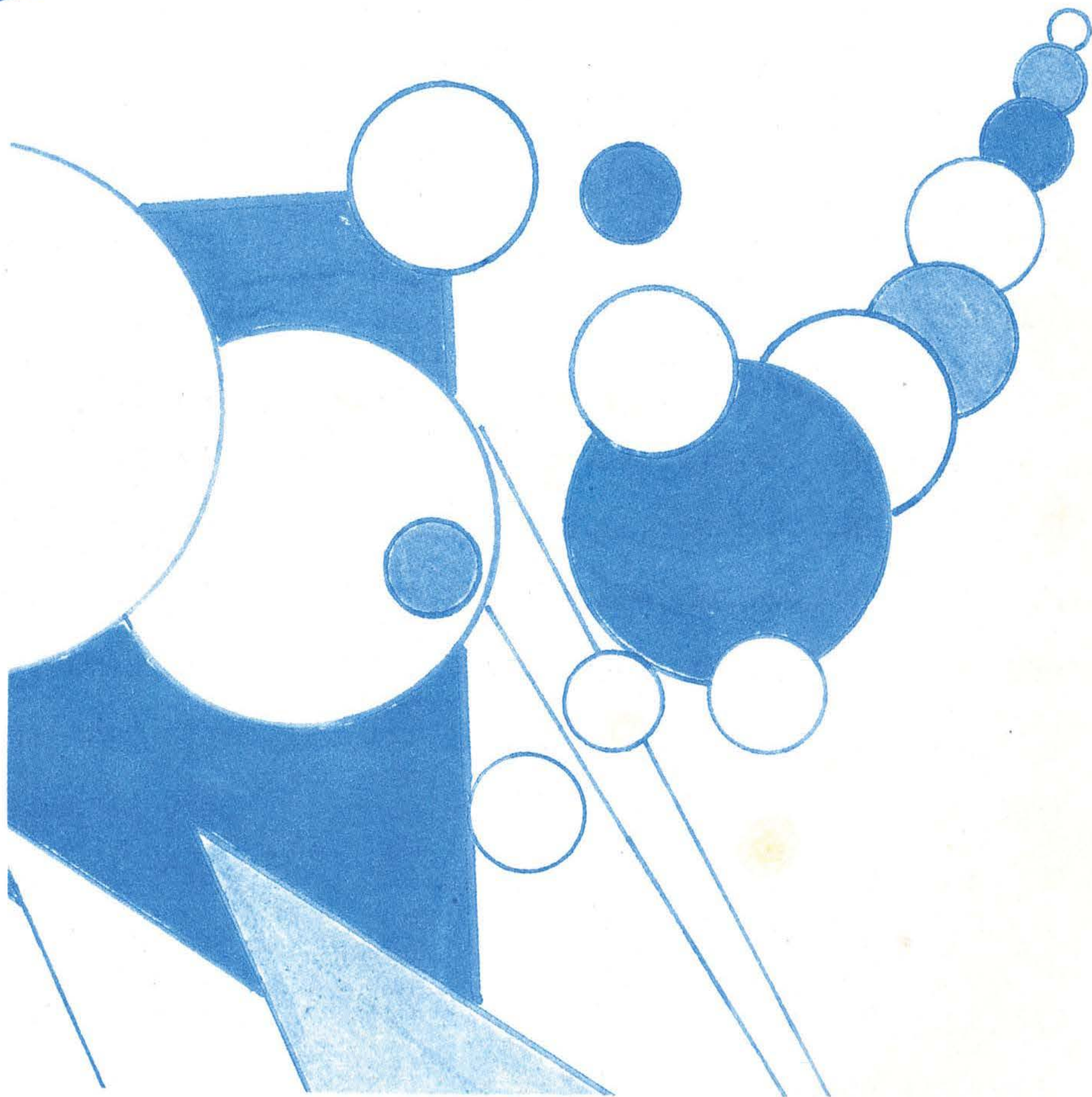
L "128.S80#1",1
L "128.S80#2",1
L "128.S80#3",1
L "128.S80#4",1

A questo punto occorre eseguire un salvataggio intermedio del programma con:

S "TEMP",8,02801,04009

Ora digitate X per rientrare in ambiente BASIC, e passate in modo 64 con GO64. Dal modo 64 digitate le istruzioni seguenti:

LOAD"128.S80#5",1,
NEW
LOAD"TEMP",8
SYS49152



Inserite quindi il disco su cui desiderate salvare la versione definitiva di SpeedScript-80 e digitate:

SAVE"SPEEDSCRIPT-80",8

A questo punto potete cancellare dal disco il file "TEMP" prima generato.

Come utilizzare SpeedScript-80

Per utilizzare SpeedScript-80 seguite questa procedura:

1. accendete il vostro C128 in modo 128
2. digitate GO64 per passare in modo 64
3. caricate normalmente SPEEDSCRIPT 80, il quale andrà attivato con un normale RUN.
4. modificate la visualizzazione dello schermo dalle 40 alle 80 colonne.

Se la procedura sopra elencata non viene eseguita correttamente, probabilmente il programma non funzionerà.

Se infatti si tiene premuto il tasto Commodore per entrare rapidamente in modo 64 all'accensione della macchina, il chip a 80 colonne non viene inizializzato e il set di caratteri non viene caricato nella memoria a 80 colonne.

Alcune modifiche nei comandi

Tre comandi di SpeedScript 3.0 non sono più disponibili utilizzando SpeedScript-80: CTRL+L (colore testo),

CTRL+B (colore fondo) e CTRL+X (inversione di due caratteri). Il comando CTRL+4, per elencare il contenuto della directory, funziona in modo analogo alla versione su 40 colonne, ma lista i file su due colonne, essendoci più spazio disponibile a video (si possono visualizzare al massimo 50 nomi di file in una singola schermata); per il modo in cui è gestito il video vi è inoltre un piccolo problema nella visualizzazione del numero di blocchi occupati dai file su disco, che ricompaiono dalla parte opposta dello schermo.

Per il resto, SpeedScript-80 è completamente compatibile con SpeedScript 3.0, anche per quanto riguarda il caricamento dei testi generati dalla versione 3.0.

Note per la digitazione dei programmi

Gli abbonati che non dispongono della cassetta allegata alla rivista devono utilizzare MLX (vedi nelle pagine fisse) per digitare i cinque programmi listati in fondo all'articolo. La procedura è la seguente:

1. Spegner e riaccendere il C128, passare in modo 64 con GO64 e digitare in modo diretto (senza numero di linea) le seguenti istruzioni:

POKE44,53:POKE53*256,0:NEW

2. Caricare e mandare in esecuzione MLX, rispondendo alle sue due domande iniziali con:

indirizzo iniziale: 10398
indirizzo finale: 10594

3. Digitare il Programma 1 e salvarlo su nastro o disco.

4. Ridare il RUN a MLX (senza spegnere o resettare il computer), dando questa volta i seguenti indirizzi:

indirizzo iniziale: 10830
indirizzo finale: 10845

5. Digitare il Programma 2 e salvarlo su nastro o disco.

6. Ripetere la procedura dei punti 4 e 5 per i Programmi 3, 4 e 5, i quali hanno rispettivamente gli indirizzi:

Programma 3:

indirizzo iniziale: 12637
indirizzo finale: 12708

Programma 4:

indirizzo iniziale: 13381
indirizzo finale: 13420

Programma 5:

indirizzo iniziale: 49152
indirizzo finale: 49463

Dopo aver digitato e salvato tutti e cinque i programmi, seguire le istruzioni date più sopra per la generazione di SpeedScript-80.

SpeedScript-80 - Programma 1

```
10398 :076,182,008,142,000,214,012
10404 :208,007,072,169,031,141,024
10410 :000,214,104,044,000,214,234
10416 :016,251,141,001,214,096,127
10422 :169,000,162,018,032,161,212
10428 :008,232,169,080,032,161,102
10434 :008,173,017,032,133,251,040
10440 :173,018,032,133,252,162,202
10446 :001,160,000,177,251,153,180
10452 :060,003,200,041,127,201,076
10458 :031,240,019,192,080,208,220
10464 :240,136,177,251,041,127,172
10470 :201,032,240,005,136,208,028
10476 :245,160,079,200,132,059,087
10482 :160,000,185,060,003,032,170
10488 :166,008,200,196,059,208,061
10494 :245,024,152,101,251,133,136
10500 :251,165,252,105,000,133,142
10506 :252,224,001,208,003,140,070
```

```
10512 :016,032,032,040,009,232,121
10518 :224,025,240,003,076,207,029
10524 :008,165,251,141,027,032,140
10530 :165,252,141,028,032,096,236
10536 :192,080,240,008,169,032,249
10542 :032,166,008,200,208,244,136
10548 :096,000,064,064,001,000,021
10554 :000,004,000,004,004,004,074
10560 :255,255,255,255,255,255,058
10566 :255,255,255,255,255,255,064
10572 :255,255,255,255,255,255,070
10578 :255,255,255,255,255,255,076
10584 :255,255,255,255,255,255,082
10590 :255,247,065,065,065,049,072
```

Programma 2

```
10830 :169,000,168,032,150,017,102
10836 :032,040,009,169,000,076,154
10842 :150,017,255,255,048,048,095
```


Programma 5

49152	: 169,018,133,251,169,030,002
49158	: 133,252,160,000,177,251,211
49164	: 041,127,240,020,201,032,161
49170	: 176,002,169,042,201,064,160
49176	: 144,010,056,233,064,201,220
49182	: 032,144,003,024,105,032,114
49188	: 145,251,200,208,225,166,207
49194	: 252,224,032,240,005,232,003
49200	: 134,252,208,214,160,000,248
49206	: 132,251,169,008,133,252,231
49212	: 177,251,201,032,240,015,208
49218	: 200,208,247,166,252,232,091
49224	: 134,252,224,021,208,238,125
49230	: 076,118,192,132,002,132,218
49236	: 253,165,252,133,254,160,021
49242	: 001,177,253,201,210,208,116
49248	: 016,200,177,253,201,255,174
49254	: 208,009,169,017,145,253,135
49260	: 136,169,128,145,253,164,079
49266	: 002,076,066,192,169,079,186

- DIECI LEZIONI (fascicoli con cassetta) SOFTWARE PER C64/128 E C64 PERSONAL COMPUTER
- DA RILEGARE IN UNO SPLENDIDO VOLUME
- IN EDICOLA OGNI 15 GIORNI A L. 8.000

ESPRIMI IL

La prima grande opera a fascicoli con cassetta software per imparare in modo interattivo i segreti del millenario gioco degli scacchi.

A SCUOLA DI SCACCHI

STORIA, TEORIA E PRATICA
PER PRINCIPIANTI ED ESPERTI

Chi sono veramente Karpov e Kasparov?
Come è nato e come si è diffuso nei secoli questo nobile gioco?
Come riuscire a vincere tutte le partite?
Scopri tutti i segreti dei grandi campioni, le loro mosse più abili e famose e le strategie di gioco.
Un'opera rivoluzionaria da leggere, da consultare, da giocare.

**IN EDICOLA
DUE MOSSE VIN**

CON LA 1^a LEZIONE
LA FAVOLOSA
SCACCHIERA
ELETTRONICA



49272 :141,227,020,169,020,141,070
 49278 :228,020,169,006,141,250,172
 49284 :020,169,166,141,252,020,132
 49290 :169,008,141,253,020,169,130
 49296 :032,141,180,022,141,133,025
 49302 :024,169,069,141,181,022,244
 49308 :169,020,141,182,022,169,091
 49314 :093,141,134,024,169,017,228
 49320 :141,135,024,169,166,141,176
 49326 :124,009,169,008,141,125,238
 49332 :009,160,004,169,120,153,027
 49338 :139,014,153,010,022,169,181
 49344 :088,153,155,014,153,026,013
 49350 :022,136,169,234,153,139,027
 49356 :014,153,010,022,153,155,199
 49362 :014,153,026,022,136,016,065

49368 :241,169,079,141,184,029,035
 49374 :141,006,030,169,020,141,217
 49380 :185,029,141,007,030,169,021
 49386 :034,141,140,027,169,022,255
 49392 :141,141,027,160,011,185,137
 49398 :028,193,153,036,022,136,046
 49404 :016,247,160,008,185,040,140
 49410 :193,153,229,020,136,016,237
 49416 :247,169,234,141,249,020,044
 49422 :141,250,020,169,040,141,007
 49428 :252,020,169,009,141,253,096
 49434 :020,096,185,069,032,240,156
 49440 :006,032,210,255,200,208,175
 49446 :245,096,169,054,133,001,224
 49452 :152,024,105,067,168,000,048
 49458 :000,000,000,000,000,000,050

TUO TALENTO

Per esaltare le tue capacità artistiche per imparare a capire tutte le applicazioni possibili di grafica con il computer.



STRUMENTI, APPLICAZIONI E IMMAGINI COL COMPUTER

Troverai tutti i consigli, i trucchi, i suggerimenti e gli aiuti necessari per sfruttare le potenzialità grafiche del computer in modo nuovo e originale. Dall'architettura all'abbigliamento, dalle auto al mondo dello spettacolo, non c'è settore in cui la computer graphics non sia applicata, prova anche tu con il tuo Commodore.



STRUMENTI, APPLICAZIONI E IMMAGINI COL COMPUTER

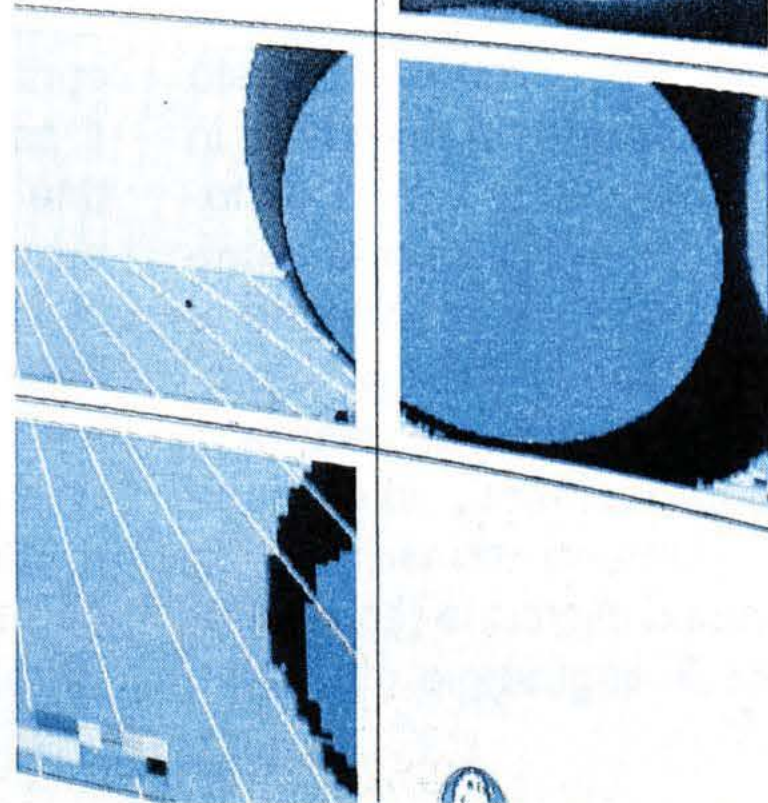
COMMODORE C64/128 C64 PERSONAL COMPUTER

APPLICAZIONI

COMPUTER STYLE
VIDEO
ARCHITETTURA
E COMPUTER GRAFICA
JOYSTICK

PROGRAMMAZIONE

COSA FARE - COME FARE - SI PUO' FARE
I PRIMI PASSI DELLA COMUNICAZIONE
ELEMENTI DI BASE
GLI STRUMENTI: SCREEN EDITOR
VIDEO E CHIP
COME FARE CON IL BASIC
DUE OCCHI, UNA BOCCA



- DIECI VIDEOLEZIONI (fascicoli con cassetta) SOFTWARE PER C64/128 E C64 PERSONAL COMPUTER
- DA RILEGARE IN UNO SPLENDIDO VOLUME
- IN EDICOLA OGNI 15 GIORNI A L. 8.000



GRUPPO EDITORIALE
JACKSON
DIVISIONE GRANDI OPERE

CENTI JACKSON

WORK

Speedy strings

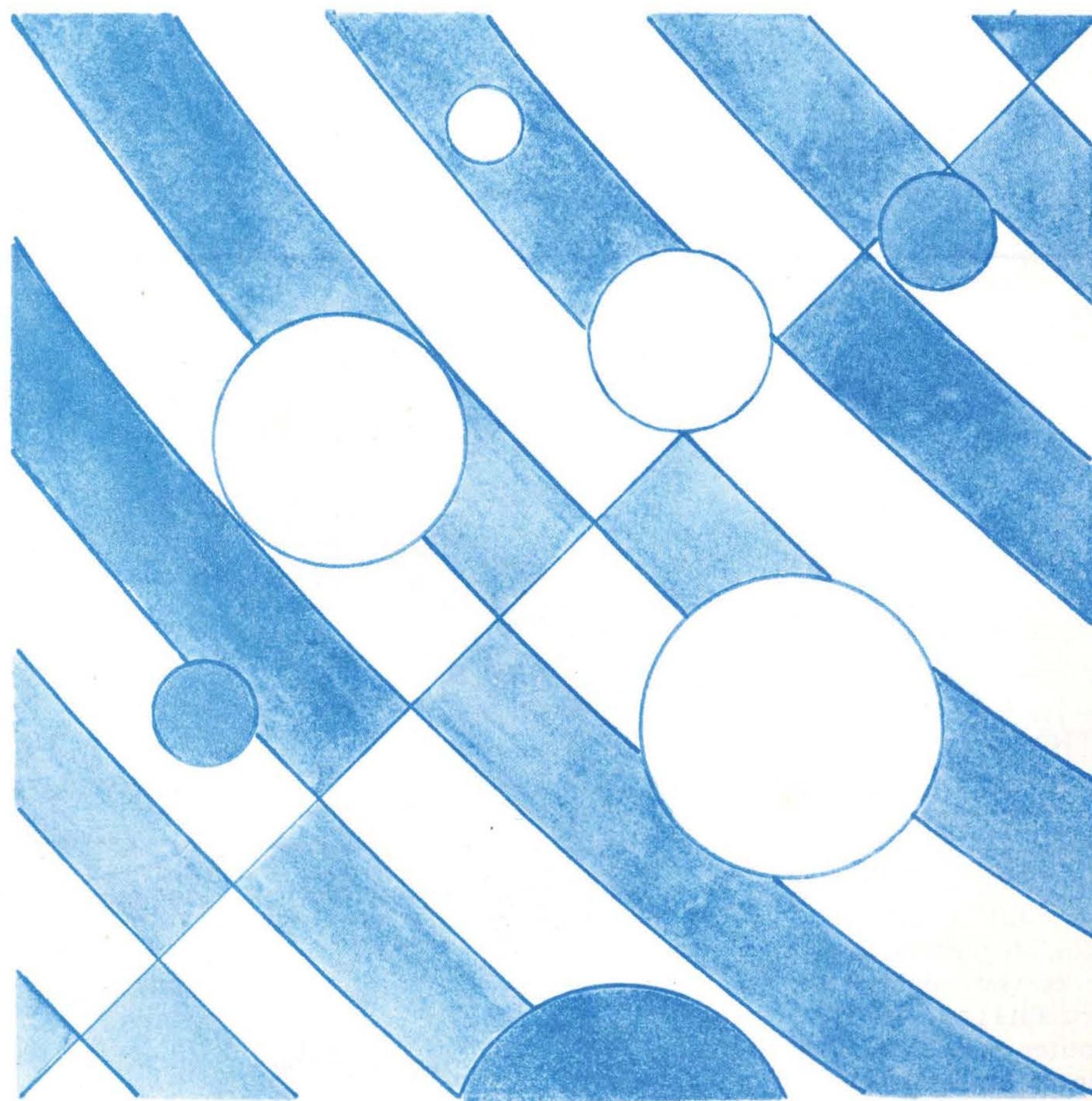
di T. Friedman
trad. e adatt. di S. Colombo

Dopo aver pubblicato SpeedFile, il "turbo" che agisce anche sui file sequenziali su nastro, ecco ora una interessante utility per i possessori di disk drive che permette di velocizzare la lettura dei file sequenziali su disco.

Nel BASIC Commodore i due sistemi convenzionali per la lettura di dati dal disco prevedono l'utilizzo delle istruzioni GET# e INPUT# (per una completa trattazione di queste istruzioni fate riferimento all'articolo "File sequenziali: quanto li conosciamo?" recentemente pubblicato). Benchè l'istruzione GET# sia la più flessibile, essa è molto più lenta della INPUT# che legge un'intera stringa piuttosto che un carattere alla volta. Anche utilizzando la INPUT#, la lettura di file sequenziali piuttosto estesi può rivelarsi un processo di estenuante lentezza.

Speedy Strings offre un'alternativa più veloce che potrete trovare utile in una gran varietà di applicazioni. Il principio su cui funziona la routine è il seguente: si crea innanzi tutto un vettore di stringa in memoria, dotato di elementi della stessa lunghezza; si caricano quindi da disco i dati del vettore, attraverso una veloce routine in linguaggio macchina, sistemandoli direttamente al posto degli elementi prestabiliti del vettore.

Non preoccupatevi, se vi sembra complicato: vedremo degli esempi che mostreranno chiaramente quanto tempo que-



sta tecnica può far risparmiare.

Per prima cosa salvate su disco i due programmi presentati in fondo a questo articolo, quindi caricate e mandate in esecuzione il Programma 1, che dimostra la differenza in velocità tra la tecnica abituale di lettura delle stringhe e quella veloce di Speedy Strings. Il programma inizia con il creare un vettore di stringa con 600 elementi, che viene riempito con elementi consistenti in stringhe di 20 spazi ciascuna (linee 100-120). La routine in linguaggio macchina

viene quindi inserita in memoria (linea 130) e viene creato un file sequenziale contenente i dati (linea 140).

Il programma chiama quindi una subroutine di rilettura dei dati con il metodo tradizionale, utilizzando una INPUT# in un ciclo FOR/NEXT, e visualizza il tempo impiegato per la lettura dell'intero file. Al termine di quest'ultima il file viene riletto utilizzando la tecnica di Speedy Strings e, come potrete notare dalla differenza dei tempi, in modo molto più veloce.

Prima di mandare in esecuzione la routi-

ne in linguaggio macchina per la lettura veloce del file dovete aver creato in memoria un vettore già pronto per contenere i dati, vettore caratterizzato da elementi di uguale lunghezza. Creando questo vettore dovete assicurarvi che la stringa costituente ciascun elemento inizi con uno spazio - CHR\$(32) - come mostrato nella linea 120 del Programma 1. In caso contrario la routine in linguaggio macchina non funzionerà correttamente.

Il Programma 2 (Fast Disk Menu) dimostra un'applicazione pratica di questa tecnica: permette infatti di leggere velocemente la directory di un disco, di ordinarla alfabeticamente e di caricare qualsiasi programma contenuto sul disco stesso.

Quando gli date il RUN, il contenuto del disco viene listato sul video in modo analogo al risultato di un LOAD"\$",8 seguito da un LIST; i file non programma (PRG) sono evidenziati da un colore diverso, mentre alla sinistra di ciascun nome di file si trova un numero progressivo. Se il disco contiene più programmi di quanti possano essere visualizzati in una singola schermata, potete premere la barra spaziatrice per visualizzare i rimanenti.

Per caricare e mandare in esecuzione un programma tra quelli elencati premete il tasto F1 seguito dal numero relativo al programma desiderato: quest'ultimo verrà caricato e mandato in esecuzione con un RUN. Si noti che Fast Disk Menu

consente di caricare e mandare in esecuzione soltanto programmi che siano attivabili con un RUN, e non quelli attivabili con una SYS.

Potete inoltre stampare su carta il contenuto della directory premendo F5, oppure fare in modo che i file vengano elencati alfabeticamente attraverso la pressione di F3.

La routine di velocizzazione può facilmente essere inserita nei propri programmi, aggiungendovi le linee DATA del Programma 1 (620-690) e la linea 610, che provvede a leggerle e a inserirle in memoria nel buffer del registratore. La lettura del file avverrà con una sintassi identica a quella della linea 210 del Programma 1.

Speedy Strings

Programma 1

```

100 MM=600:REM MAX. MEMORIA :rem 58
110 DIMF$(MM):PRINT"{CLR} INIZIALIZZAZIONE VETTORE..." :rem 229
120 FORI=0TOMM:F$(I)=CHR$(32)+"{ 19 SPAZI}":NEXT:REM* 19 SPAZI=LEN 20 :rem 62
130 GOSUB600:REM CARICAMENTO L/M:rem 182
140 GOSUB300:REM CREA FILE SEQUENZIALE SU DISCO :rem 97
150 GOSUB400:REM LETTURA CONVENZIONALE :rem 196
160 GOSUB200:REM LETTURA VELOCE :rem 166
170 END :rem 111
200 PRINT"{ 2 GIU'} LETTURA VELOCE" :rem 100
210 TI$="000000":OPEN15,8,15:OPEN1,8,0,"0":TEST 0,S,R":SYS(832):CLOSE1:CLOSE15 :rem 127
220 C=PEEK(980)+256*PEEK(981) :rem 53
230 T$=TI$ :rem 11
240 FORI=0TOC-1:PRINTI;F$(I):NEXT :rem 63
250 PRINT,MID$(T$,3,2)": "RIGHT$(T$,2) :rem 27
260 RETURN :rem 120
290 REM CREA IL FILE SEQUENZIALE:rem 149
300 PRINT"{ 2 GIU'}CREAZIONE FILE SEQ SU DISCO PER TEST" :rem 112
310 FORI=0TOMM:F$(I)="*1234567890123456789":NEXT:REM *20 CARATTERI* :rem 45
320 OPEN2,8,2,"@0:TEST 0,S,W" :rem 71
330 FORI=0TOMM:PRINT#2,F$(I):NEXT:CLOSE2:RETURN :rem 49
400 PRINT"{ 2 GIU'} LETTURA CONVENZIONALE" :rem 131
410 TI$="000000":OPEN15,8,15:OPEN2,8,0,"0":TEST 0,S,R":C=0 :rem 78
420 IFST=64THEN440 :rem 58
430 INPUT#2,F$(C):C=C+1:GOTO420 :rem 8
440 CLOSE2:CLOSE15:T$=TI$ :rem 6

```

```

450 FORI=0TOC-1:PRINTI;F$(I):NEXT :rem 66
460 PRINT,MID$(T$,3,2)": "RIGHT$(T$,2) :rem 30
480 RETURN :rem 124
590 REM *ROUTINE UTILIZZATA PER LA LETTURA RAPIDA DEL FILE SEQUENZIALE* :rem 142
600 PRINT"{ 2 GIU'} MEMORIZZAZIONE L/M" :rem 148
610 AD=832:FORI=0TO79:READD:POKEI+AD,D:NEXT:RETURN :rem 223
620 DATA169,255,141,212,3,141,213,3,165,55 :rem 61
630 DATA 133,252,165,56,133,253,162,1,32,198,255 :rem 64
640 DATA 174,212,3,232,142,212,3,208,3,238,213,3,216,56,165,252,233 :rem 210
650 DATA 21:REM LUNGHEZZA STRINGHE +1 :rem 174
660 DATA 133,252,176,5,166,253,202,134,253,32,207,255 :rem 52
670 DATA 164,144,208,18,201,13,240,245,160,0,145,252,200,192 :rem 122
680 DATA20:REM LUNGHEZZA STRINGHE :rem 84
690 DATA 240,209,32,207,255,208,244,32,204,255,96,0,0,0 :rem 138

```

Programma 2

```

100 POKE53280,6:POKE53281,6 :rem 242
110 PRINTCHR$(14)CHR$(8):PRINT"{CLR}{BLK}{ 10 GIU'}{YEL}"SPC(12)"{ 1 SH F}AST{ 1 SH D}ISK { 1 SH M}ENU :rem 160
130 GOSUB570:QQ=AD :rem 78
140 DIMF$(120):FORI=0TO120 :rem 202
150 F$(I)=CHR$(32)+"{ 20 SPAZI}":NEXT:Y$=CHR$(34):Z$=CHR$(190) :rem 4
160 OPEN1,8,0,"$":SYSQQ:CLOSE1 :rem 10
170 C=PEEK(0):A=1 :rem 167

```



```

180 IFA=>CTHENA=1 :rem 5
190 PRINT"{CLR}{GIU'}{ 7 SPAZI}{RVS}[<8>]
  "Y$LEFT$(F$(0),16)Y$"{ 3 SPAZI}"MID$(
  F$(0),18,2) :rem 69
200 PRINT:FORI=ATO+8:IFI=>CTHENPRINT:GOT
  O270 :rem 230
210 IFLEFT$(F$(I),1)=Z$THENE=2:GOTO230
  :rem 201
220 E=1:IFMID$(F$(I),18,3)<>"PRG"THENE=2:
  F$(I)=Z$+LEFT$(F$(I),20) :rem 55
230 IFE=2THENPOKE646,3:GOTO250 :rem 63
240 POKE646,7 :rem 200
250 PRINTTAB(5)I;Y$MID$(F$(I),E,16)Y$;
  :rem 145
260 PRINT"{ 3 SPAZI}"MID$(F$(I),17+E,3)
  :rem 31
270 PRINT:NEXT :rem 159
280 PRINT"{GIU'}[<8>]{ 5 SPAZI}{ 1 SH F}1
  -{ 1 SH L}{ 1 SH O}{ 1 SH A}{ 1 SH D}
  { 3 SPAZI}{ 1 SH F}3-{ 1 SH O}
  { 1 SH R}{ 1 SH D}{ 1 SH I}{ 1 SH N}
  { 1 SH A}{ 2 SPAZI}{ 1 SH F}5-
  { 1 SH S}{ 1 SH T}{ 1 SH A}{ 1 SH M}
  { 1 SH P}{ 1 SH A}":POKE198,0:WAIT198
  ,1 :rem 1
290 IFPEEK(197)=4THEN340 :rem 120
300 IFPEEK(197)=5THEN390 :rem 118
310 IFPEEK(197)=6THEN460 :rem 118
320 IFPEEK(197)=60THENA=A+9:GOTO180
  :rem 60
330 PRINT"{ 3 SU}":GOTO280 :rem 39
340 INPUT"{GIU'}{ 3 SPAZI}{ 1 SH P}ROGRAM
  MA #";N$:N=VAL(N$):IFN<1ORN>CTHEN190
  :rem 154
350 F$=LEFT$(F$(N),16) :rem 123
360 IFRIGHT$(F$,1)=" "THENF$=LEFT$(F$,LEN
  (F$)-1):GOTO360 :rem 115
370 PRINT"{CLR}LOAD"Y$F$Y$",8 :rem 8
380 POKE631,19:POKE632,13:POKE633,82:POKE
  634,117:POKE635,13:POKE198,5:END
  :rem 8
390 PRINT"{CLR}{ 5 GIU'}{ 4 SPAZI}
  { 1 SH O}{ 1 SH R}{ 1 SH D}{ 1 SH I}
  { 1 SH N}{ 1 SH A}{ 1 SH M}{ 1 SH E}
  { 1 SH N}{ 1 SH T}{ 1 SH O}{ 1 SH I}
  { 1 SH N}{ 1 SH E}{ 1 SH S}{ 1 SH E}
  { 1 SH C}{ 1 SH U}{ 1 SH Z}{ 1 SH I}
  { 1 SH O}{ 1 SH N}{ 1 SH E}..."
  :rem 48
400 FORI=1TOC-2:IFF$(I)<F$(I+1)THEN450
  :rem 168
410 Q$=F$(I+1) :rem 167
420 FORJ=1TO1STEP-1:IFF$(J)<Q$THENF$(J+1)
  =Q$:GOTO450 :rem 136
430 F$(J+1)=F$(J):NEXT :rem 179
440 F$(1)=Q$ :rem 54
450 NEXTI:A=1:GOTO190 :rem 23
460 PRINT"{CLR}{GIU'}{ 1 SH S}TAMPANTE
  { 1 SH O}{ 1 SH K}?" :POKE198,0:WAIT19
  8,1:GETMT$:IFMT$<>"S"THEN190:rem 49
470 OPEN4,4,7 :rem 196
480 PRINT#4,Y$LEFT$(F$(0),16)Y$"
  { 3 SPAZI}"MID$(F$(0),18,2):PRINT#4,C
  HR$(20)CHR$(15); :rem 38

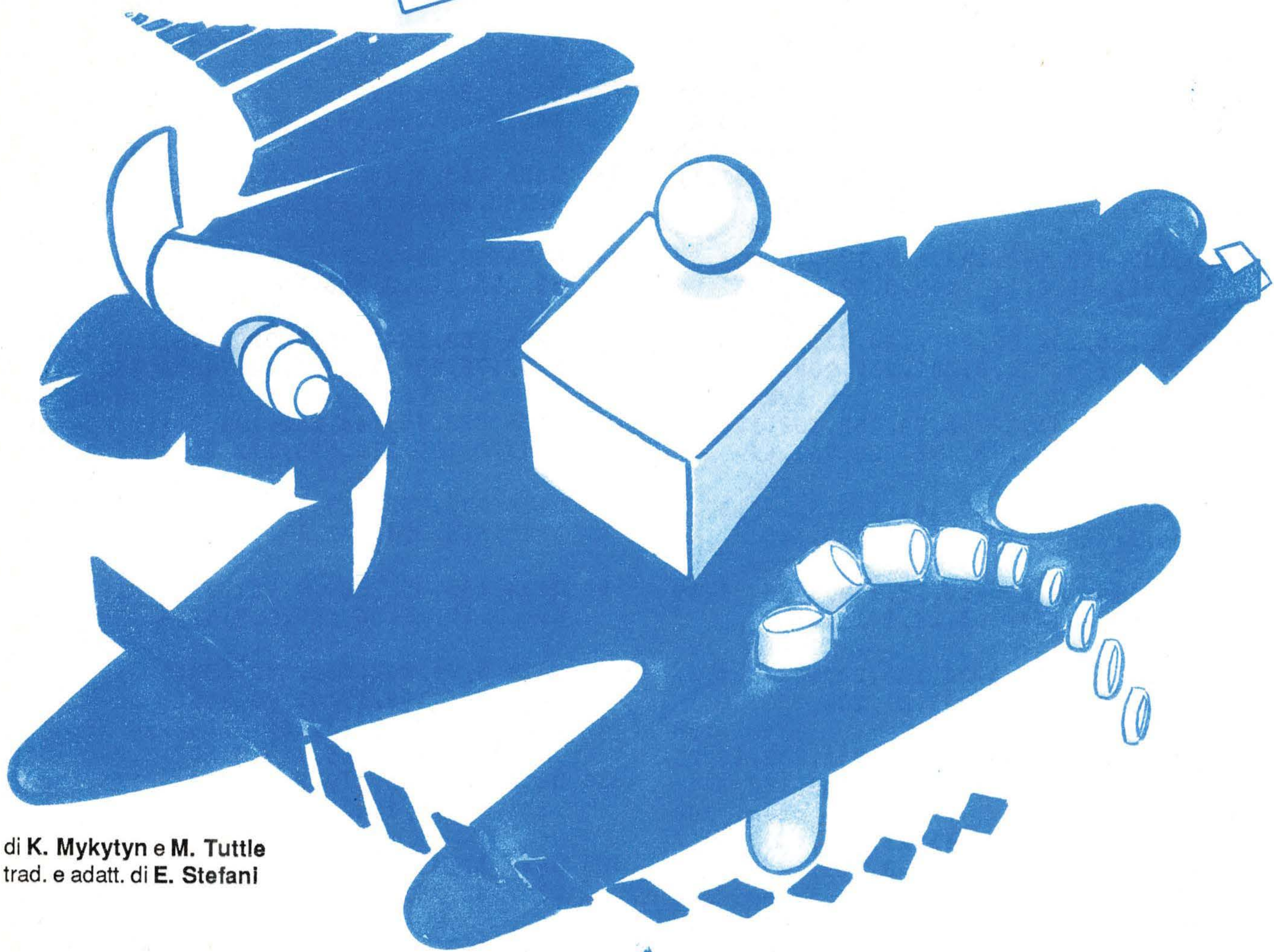
```

```

490 U=C-1:V=INT(U/3)+1 :rem 236
500 FORI=1TOV:FORJ=0TO3:Q=I+J*V:IFQ>UTHEN
  530 :rem 84
510 E=1:IFLEFT$(F$(Q),1)=Z$THENE=2
  :rem 185
520 PRINT#4,Y$MID$(F$(Q),E,16)Y$;:IFE=2TH
  ENPRINT#4,Z$; :rem 176
530 PRINT#4,"{ 5 SPAZI}";:NEXT:PRINT#4
  :rem 190
540 NEXT:CLOSE4:A=1:GOTO190 :rem 178
550 END :rem 113
560 REM MEMORIZZAZIONE LINGUAGGIO MACCHIN
  A :rem 220
570 AD=830:FORI=0TO72:READD:POKEI+AD,D:NE
  XT:RETURN :rem 219
580 DATA 169,255,133,0,165,55,133,71,165,
  56,133,72,162,1,32,198,255,166,0,232
  :rem 219
590 DATA 134,0,216,56,165,71,233,22,133,7
  1,176,5,166,72,202,134,72,32,207,255
  :rem 207
600 DATA 164,144,208,22,201,34,208,245,16
  0,0,32,207,255,201,34,240,249,145,71
  :rem 187
610 DATA 200,192,21,208,242,240,207,32,20
  4,255,96,0,0,0 :rem 117

```


Arcade baseball



di K. Mykytyn e M. Tuttle
trad. e adatt. di E. Stefani

Ecco la versione per C64 di un popolare gioco elettronico commerciale che vi renderà provetti giocatori del famoso sport americano.

Dopo una breve attesa iniziale il programma vi richiede di premere il tasto F1 per impratichirvi nel gioco, oppure il tasto F7 per iniziare una partita. In quest'ultimo caso è possibile selezionare una partita contro il computer (1 giocatore) oppure contro un altro avversario (2 giocatori).

Se premete F1, entrate in "modo pratica": vi vengono lanciate 20 palle (un misto casuale di lanci veloci, con effetto, curvi e diretti) che dovete cercare di respingere premendo la barra spaziatrice nel momento giusto.

Nel gioco reale, invece, lo schermo è diviso in tre parti: a sinistra si trova il campo di gioco, nella cui parte alta si trova una fila di buche nere. Una palla battuta finirà in una di queste buche, ciascuna delle quali è contrassegnata da un punteggio (singolo, doppio, triplo, oppure "out"). Per ottenere un cosiddetto "ho-

me run" la palla ribattuta deve passare sopra il centro del rettangolo rosso nella parte mediana del campo di gioco.

Nell'angolo superiore destro dello schermo vengono visualizzati i punteggi e i dati relativi alla partita in corso, e i giocatori sono rappresentati come la squadra di casa e i visitatori (home e visitors, all'americana).

Nel rettangolo sottostante è visualizzato il "diamante", che mostra quali basi sono occupate dalla squadra al momento del lancio.

Se si gioca contro il computer, la partita

dura per tre "out", prima dei quali occorre cercare di totalizzare il massimo punteggio; il computer seleziona a caso il tipo di lancio, al quale dovete rispondere premendo la barra spaziatrice. Vi accorgete che è molto importante scegliere il momento adatto per compiere questa operazione e colpire la palla in arrivo, pena il regalo di uno "strike" all'avversario. La "mazza" che muovete attraverso la

barra spaziatrice è di forma semicircolare, in modo da dare traiettorie diverse alla palla, in funzione del punto in cui questa colpisce la mazza stessa. La partita con due giocatori avviene alternativamente con un giocatore che controlla la mazza attraverso la barra spaziatrice e l'altro che lancia la palla premendo uno dei tasti funzione. I tiri possibili sono:

F1	tiro veloce
F3	con cambio di traiettoria
F5	con effetto
F7	tiro curvo

Con due giocatori la partita dura tre "inning", la squadra di casa ha colore blu, mentre i visitatori hanno colore rosso. Buon divertimento!

Arcade Baseball

```

10 POKE56,48:POKE55,0:CLR:GOSUB780
                                :rem 54
20 PRINT"{ 7 GIU' }"SPC(14)CHR$(14)"[<4>]
   { 1 SH A}TTENDERE...{ 8 SU}" :rem 6
30 FORA=49152TO49999:READB:POKEA,B:C=C+B:
   NEXT :rem 27
40 IFC<>104062THENPRINT"{CLR}{ 1 SH E}RRO
   RE NELLE ISTRUZIONI { 1 SH D}{ 1 SH A}
   { 1 SH T}{ 1 SH A}!":END :rem 155
50 DATA 173,4,220,141,166,3,169,18,141,5,
   212,169,100,141,1,212,169,0,133
                                :rem 189
60 DATA 187,133,188,133,189,173,30,208,32
   ,199,192,169,0,141,167,3,32,85,193
                                :rem 139
70 DATA 32,15,194,32,254,194,32,182,194,3
   2,15,194,173,120,3,201,250,208,8
                                :rem 4
80 DATA 169,0,141,162,3,76,152,192,174,16
   7,3,208,114,201,76,176,220,201,69
                                :rem 62
90 DATA 144,216,169,27,141,163,3,169,37,1
   41,164,3,162,7,173,100,3,205,163
                                :rem 7
100 DATA 3,144,8,205,164,3,176,3,76,38,19
   2,173,163,3,24,105,24,141,163
                                :rem 107
110 DATA 3,173,164,3,24,105,24,141,164,3,
   202,16,219,169,2,141,21,208,173
                                :rem 189
120 DATA 100,3,56,233,9,162,255,232,56,23
   3,24,176,250,189,175,192,141,162,3
                                :rem 108
130 DATA 120,169,240,141,26,208,169,49,14
   1,20,3,169,234,141,21,3,169,129,141
                                :rem 150
140 DATA 13,220,88,96,1,5,2,5,3,5,2,5,1,2
   01,30,240,3,76,38 :rem 2
150 DATA 192,169,4,141,162,3,76,152,192,1
   69,2,141,21,208,169,1,141,28,208
                                :rem 11
160 DATA 160,7,169,0,153,110,3,136,16,248
   ,169,90,141,101,3,169,230,141,121
                                :rem 39
170 DATA 3,169,227,141,131,3,169,0,141,40
   ,208,169,12,141,37,208,169,11,141
                                :rem 47
180 DATA 38,208,169,127,141,13,220,169,21
   4,141,20,3,169,193,141,21,3,169,27
                                :rem 108
190 DATA 141,17,208,169,129,141,26,208,16
   9,250,141,18,208,169,117,141,100,3,16

```

```

9 :rem 59
200 DATA 130,141,120,3,169,129,141,26,208
   ,96,165,187,208,26,169,127,141,0,220
                                :rem 201
210 DATA 169,0,141,1,220,173,1,220,201,23
   9,240,1,96,165,189,208,14,230,187
                                :rem 41
220 DATA 230,189,238,101,3,173,101,3,201,
   145,240,1,96,169,0,133,187,96,173
                                :rem 51
230 DATA 165,3,240,24,169,0,133,162,32,18
   9,193,41,63,9,64,197,162,208,252
                                :rem 20
240 DATA 32,189,193,41,3,170,76,147,193,1
   69,255,141,0,220,169,0,141,1,220
                                :rem 3
250 DATA 169,254,141,0,220,173,1,220,201,
   248,176,234,73,255,74,74,74,162,255
                                :rem 162
260 DATA 232,74,144,252,134,2,189,206,193
   ,133,5,189,210,193,133,6,189,202,193
                                :rem 221
270 DATA 133,253,169,0,133,254,169,20,133
   ,248,169,224,141,130,3,169,0,133,251
                                :rem 201
280 DATA 133,252,169,3,141,21,208,96,173,
   166,3,10,10,56,109,166,3,141,166
                                :rem 6
290 DATA 3,96,200,250,180,200,5,0,0,248,0
   ,0,0,255,169,1,141,25,208 :rem 138
300 DATA 169,0,133,3,160,7,162,14,185,100
   ,3,157,0,208,185,120,3,157,1 :rem 38
310 DATA 208,185,110,3,10,102,3,185,130,3
   ,153,248,7,202,202,136,16,227,165
                                :rem 29
320 DATA 3,141,16,208,169,250,141,18,208,
   230,162,76,188,254,198,247,240,1,96
                                :rem 173
330 DATA 169,90,133,247,32,39,193,165,253
   ,24,109,140,3,141,140,3,173,120,3
                                :rem 48
340 DATA 101,254,141,120,3,174,167,3,208,
   26,201,66,208,22,169,0,133,254,169
                                :rem 96
350 DATA 50,133,253,165,252,48,6,169,60,1
   33,251,208,4,169,190,133,251,173,150
                                :rem 212
360 DATA 3,24,101,251,141,150,3,173,100,3
   ,101,252,141,100,3,201,16,240,4
                                :rem 149
370 DATA 201,217,208,17,165,251,73,255,24
   ,105,1,133,251,165,252,73,255,105,0
                                :rem 146

```



```

380 DATA 133,252,198,248,208,60,169,20,13
    3,248,165,251,24,101,5,133,251,165,25
    2 :rem 57
390 DATA 101,6,133,252,165,254,16,21,173,
    130,3,201,224,240,6,206,130,3,76
    :rem 236
400 DATA 181,194,169,226,141,130,3,76,181
    ,194,173,130,3,201,226,240,6,238,130
    :rem 198
410 DATA 3,76,181,194,169,224,141,130,3,9
    6,165,188,208,67,173,30,208,41,3
    :rem 23
420 DATA 201,3,208,58,169,0,133,5,133,6,2
    30,188,173,100,3,56,237,101,3
    :rem 95
430 DATA 24,105,15,16,2,169,0,201,28,144,
    2,169,27,10,170,189,22,195,133
    :rem 154
440 DATA 251,189,23,195,133,252,169,128,1
    41,4,212,169,129,141,4,212,169,255,13
    3 :rem 64
450 DATA 254,169,120,133,253,96,173,120,3
    ,201,120,208,16,173,100,3,201,113,144
    :rem 225
460 DATA 9,201,121,176,5,169,1,141,167,3,
    96,64,255,87,255,100,255,113,255
    :rem 21
470 DATA 126,255,139,255,152,255,165,255,
    188,255,201,255,214,255,227,255,240
    :rem 187
480 DATA 255,0,0,0,0,16,0,29,0,42,0,55,0,
    68,0,91,0,104,0,117,0 :rem 170
490 DATA 130,0,143,0,156,0,169,0,192,0,19
    2,0 :rem 91
500 ML$=" [<I>]" +CHR$(8)+" [<X>]" +CHR$(3)+
    " [<2>]" +X{ 1 SH J} +CHR$(16)+CHR$(248)+
    "L [<B>]" + [<T>]" :POKE835,0 :rem 233
510 POKE836,208:POKE830,0:POKE831,216:POK
    E828,0:POKE829,56:POKE56334,0
    :rem 42
520 POKE1,51:ML$=ML$:SYS (PEEK (51)+256*PEE
    K (52)):POKE1,55:POKE56334,1 :rem 170
530 FORI=12568TO12671:READJ:POKEI,255-J:N
    EXT :rem 185
540 FORI=12672TO12727:READJ:POKEI,J:NEXT
    :rem 239
550 DATA 0,115,219,219,219,219,113,0,0,11
    1,102,102,102,102,198,0 :rem 37
560 DATA 149,149,149,149,165,170,170,170,
    86,86,86,86,90,170,170,170 :rem 14
570 DATA 170,170,170,165,149,149,149,149,
    170,170,170,90,86,86,86,86 :rem 15
580 DATA 130,130,130,130,130,170,170,170,
    128,0,0,0,0,0,0,0 :rem 226
590 DATA 255,255,255,126,060,024,0,0,1,0,
    0,0,0,0,0,0,0,0,0,0,0,0,0,0,1 :rem 167
600 DATA 0,0,0,0,0,0,0,128,255,255,255,12
    6,60,24,0,0,213,213,245,245,253,253
    :rem 102
610 DATA 255,255,87,87,95,95,127,127,255,
    255,125,125,125,125,125,125,125,125
    :rem 187
620 DATA 126,255,255,255,255,255,255,255,
    255,255,255,255,255,255,126
    :rem 0
630 DATA 127,255,255,255,255,255,255,127,
    254,255,255,255,255,255,255,254
    :rem 1
640 Q=14336 :rem 41
650 READB:IFB<0THENFORA=1TOABS (B):POKEQ,0
    :Q=Q+1:NEXT:GOTO650 :rem 83
660 IFB=256THENFF=1:GOTO730 :rem 125
670 POKEQ,B:Q=Q+1:GOTO650 :rem 14
680 DATA -25,60,0,0,255,0,0,85,0,0,85,0,0
    ,20,-51,20,0,0,255,0,0,255,0,0,85,0,0
    :rem 156
690 DATA 20,-51,20,0,0,85,0,0,85,0,0,255,
    0,0,60,-66,255,0,7,255,224,31,255,248
    :rem 206
700 DATA 63,255,252,127,255,254,127,255,2
    54,255,255,255,255,255,255,239,256
    :rem 148
710 DATA 129,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
    129 :rem 30
720 DATA 128,0,0,0,0,0,0,128,1,0,0,0,0,0,
    0,1 :rem 31
730 POKE53272,23:POKE53270,200 :rem 133
740 CL(1)=10:CL(2)=14:FL=0:RM=20:PL=2
    :rem 46
750 FORA=54272TO54295:POKEA,0:NEXT:POKEA,
    15:POKE54280,17:POKE54284,18:rem 113
760 IN=1:SK=0:CK=0:P1$="{RVS}HOME":P2$="V
    ISITOR":GOSUB770:GOTO860 :rem 177
770 IFFF=1THENRETURN :rem 54
780 PRINT "{CLR}{ 3 GIU' } {RED}" SPC(6);:POK
    E53281,0:POKE53280,0 :rem 137
790 PRINT "{RVS}[<V>][<C>]{OFF}{ 2 SPAZI}
    {RVS}[<K>][<D>][<F>]{OFF}[<F>][<D>]
    {RVS}[<D>][<F>]{OFF}[<F>]{ 2 SPAZI}
    {RVS}[<V>][<C>]{OFF}{ 2 SPAZI}{RVS}
    [<K>][<D>][<C>]{OFF}{ 2 SPAZI}{RVS}
    [<K>][<D>][<I>]{OFF}[<V>]" SPC(11)"
    {WHT}{RVS}[<K>][<C>][<V>]{OFF}[<K>]
    {RVS}[<K>][<C>][<V>]{OFF}[<V>]{RVS}
    [<K>]"; :rem 49
800 PRINT "{OFF}[<K>]{ 3 SPAZI}{RVS}[<K>]
    [<C>][<V>]{OFF}[<K>]{RVS}[<K>]{OFF}
    [<K>]{RVS}[<K>]{OFF}[<K>]{RVS}[<K>]
    [<C>]{OFF}[<F>]" SPC(12)" {BLU}{RVS}
    [<K>]{OFF}[<K>]{RVS}[<K>]{OFF}[<K>]
    {RVS}[<K>][<D>][<C>]{OFF}{ 2 SPAZI}
    {RVS}[<K>]{OFF}[<K>][<D>][<F>]{RVS}
    [<K>]{OFF}[<K>]{RVS}[<K>]{OFF}[<K>]";
    :rem 78
810 PRINT " {RVS}[<K>]{OFF}[<K>]{RVS}[<V>]
    {OFF}[<V>]{RVS}[<K>]{OFF}[<K>]" SPC(1
    3)" [<C>][<V>][<C>][<V>][<C>][<V>]
    [<C>][<V>]{ 2 SPAZI}{RVS}[< 2 I>]
    {OFF}{ 2 SPAZI}[<C>][<V>][<C>][<V>]
    [<C>]{RVS}[<I>]{OFF}[<V>]{ 2 SPAZI}
    [<C>]{RVS}[< 2 I>]{OFF}[<V>]
    { 2 GIU' }" SPC(9); :rem 232
820 PRINT "{RED}{RVS}[<K>][<D>][<F>]{OFF}
    [<F>]{RVS}[<V>][<C>]{OFF}[<D>]{RVS}
    [<D>][<F>]{OFF}[<F>]{RVS}[<K>][<D>]
    [<I>]{OFF}[<V>]{RVS}[<K>][<D>][<F>]
    {OFF}[<F>]{RVS}[<V>][<C>]{OFF}{RVS}
    [<K>]{OFF}[<K>]{ 2 SPAZI}{RVS}[<K>]
    {OFF}[<K>]" SPC(10)" {WHT}{RVS}[<K>]

```



```

[<C>][<V>]{OFF}[<V>]{RVS}[<K>]";
:rem 108
830 PRINT"[<C>][<V>]{OFF}[<K>][<C>]{RVS}
[<C>]{OFF}[<I>]{RVS}[<K>][<C>]{OFF}
[<F>]{RVS}[<K>][<C>][<V>]{OFF}[<V>]
{RVS}[<K>][<C>][<V>]{OFF}[<K>]{RVS}
[<K>]{OFF}[<K>]{ 2 SPAZI}{RVS}[<K>]
{OFF}[<K>]"SPC(10)"{BLU}{RVS}[<K>]
{OFF}[<K>]{RVS}[<K>]{OFF}[<K>]{RVS}
[<K>]{OFF}[<K>]{RVS}[<K>]{OFF}[<K>]
[<D>][<F>]";
:rem 113
840 PRINT"{RVS}[<K>]{OFF}[<K>]{RVS}[<K>]
{OFF}[<K>]{ 2 SPAZI}{RVS}[<K>]{OFF}
[<K>]{RVS}[<K>]{OFF}[<K>]{RVS}[<K>]
{OFF}[<K>]{RVS}[<K>]{OFF}[<K>]{RVS}
[<K>]{OFF}[<K>]{ 2 SPAZI}{RVS}[<K>]
{OFF}[<K>]"SPC(10)"[<C>]{RVS}
[< 2 I>]{OFF}[<C>][<V>][<C>][<V>]
{RVS}[< 2 I>]{OFF}[<C>]{RVS}
[< 2 I>]";
:rem 70
850 PRINT"{OFF}[<V>][<C>]{RVS}[< 2 I>]
{OFF}[<C>][<V>][<C>][<V>][<C>]{RVS}
[< 2 I>]{OFF}[<V>][<C>]{RVS}[< 2 I>]
{OFF}[<V>":RETURN
:rem 52
860 PRINTCHR$(14)"{ 4 GIU'}"SPC(12)"[<4>]
{ 1 SH P}RATICA{ 5 SPAZI}{ 1 SH F}1":
PRINTSPC(12)"{ 2 GIU'}{ 1 SH G}IOCO
{ 7 SPAZI}{ 1 SH F}7"
:rem 15
870 POKE198,0:WAIT198,1:GETMT$:IFMT$<>"
{F1}"ANDMT$<>"{F7}"THEN870
:rem 150
880 IFMT$="{F1}"THENFL=1:IN$="1":GOTO920
:rem 222
890 C$="{ 14 SPAZI}":PRINTCHR$(142)"
{ 4 SU}"SPC(12)C$:PRINT"{ 2 GIU'}"SPC
(12)C$"{ 4 SU}"
:rem 120
900 PRINTSPC(13)"{ 2 GIU'}{RVS}1{OFF} 0
{RVS}2{OFF} GIOCATORI?"
:rem 1
910 POKE198,0:WAIT198,1:GETIN$:IFIN$=""OR
IN$<"1"ORIN$>"2"THEN910
:rem 245
920 NP=VAL(IN$):POKE933,-(NP=1):PRINT"
{CLR}";:POKE53280,2:POKE53281,1:TE=6:
B$=""
:rem 245
930 POKE53272,28:POKE53282,3:POKE53283,0:
POKE53270,PEEK(53270)OR16
:rem 15
940 W=1063:POKEW,160:A$="{RVS}{OFF}"
:rem 66
950 POKEW+54272,6:C$="{RVS}{CYN}
{ 26 SPAZI}{RED}" :D$="{RVS}
{ 12 SPAZI}"
:rem 217
960 FORU=1TO2:FORI=1TO12:PRINTC$;:POKE646
,TE:PRINTD$:SM=W+40*I:POKESM,160
:rem 158
970 POKESM+54272,TE:NEXTI:TE=5:W=1543:NEX
TU:PRINTC$"{GRN}"D$"{HOME}"
:rem 34
980 PRINT"{ 11 GIU'}"SPC(27)"{RED}{RVS}
{ 12 SPAZI}{HOME}{GIU'}{WHT}":POKE154
3,160:POKE55815,2
:rem 48
990 PRINT"{RVS}{RED}1B {OFF}#{RVS} 2B
{OFF}#{RVS} 3B {OFF}#{RVS} 2B {OFF}
#{RVS} 1B"
:rem 144
1000 PRINT"[<3>]'({DES}'({DES}'({DES}'(
{DES}'({DES}'({DES}'({DES}'({DES}'(
:PRINT"%{DES}%{DES}%{DES}%{DES}%
&{DES}%{DES}%{DES}%{DES}%
:rem 109

```

```

1010 PRINT"{HOME}{ 8 GIU'}{RED}"SPC(12)"
{RVS}{ 2 SPAZI}{GIU'}{ 2 SIN}{RVS}
{ 2 SPAZI}{GIU'}{ 2 SIN}{ 2 SPAZI}":
PRINT"{ 2 GIU'}[<6>]{ 4 GIU'}[<3>]0"
SPC(24)"1":Y=24
:rem 250
1020 X=1:FORI=1TO6:FORZ=1TOX:B$=B$+A$:NEX
TZ:PRINTB$"0";:PRINTTAB(Y)"1"B$:B$="
"
:rem 228
1030 Y=Y-1:X=X+1:NEXTI:PRINT"{RVS}
{ 7 SPAZI}{OFF}0"SPC(10)"1{RVS}
{ 7 SPAZI}{HOME}{ 14 GIU'}{WHT}";
:rem 149
1040 IFFL=1THENPRINT"{HOME}{ 2 GIU'}"SPC(
29)"{RVS}{BLU}P{GIU'}R{GIU'}A{GIU'}T
{GIU'}I{GIU'}C{GIU'}A{GIU'}":GOTO11
00
:rem 80
1050 PRINTSPC(32)"{OFF}{GRN}- .":PRINTSPC
(31)"-{RVS}{ 1 SH N}{OFF}3{RVS}
{ 1 SH M}{OFF}.":PRINTSPC(30)"-{RVS}
{ 1 SH N}{OFF}*{RVS}{OFF},{RVS}
{ 1 SH M}{OFF}."
:rem 50
1060 PRINTSPC(29)"-{RVS}{ 1 SH N}{OFF}*
{RVS}{ 3 SPAZI}{OFF},{RVS}{ 1 SH M}
{OFF}.":PRINTSPC(28)"-{RVS}{ 1 SH N}
{OFF}*{RVS}{ 5 SPAZI}{OFF},{RVS}
{ 1 SH M}{OFF}."
:rem 86
1070 PRINTSPC(28)"{WHT}{GRN}5"SPC(3)"
{RVS}{ 1 SH C}"SPC(3)"{OFF}6{WHT}
{SIN}{GRN}":PRINTSPC(28)",{RVS}
{ 1 SH M}{OFF}.{RVS}{ 5 SPAZI}{OFF}-
{RVS}{ 1 SH N}{OFF}*
:rem 47
1080 PRINTSPC(29)",{RVS}{ 1 SH M}{OFF}.
{RVS}{ 3 SPAZI}{OFF}-{RVS}{ 1 SH N}
{OFF}*":PRINTSPC(30)",{RVS}{ 1 SH M}
{OFF}.{RVS}{OFF}-{RVS}{ 1 SH N}
{OFF}*
:rem 81
1090 PRINTSPC(31)",{RVS}{ 1 SH M}{OFF}4
{RVS}{ 1 SH N}{OFF}*":PRINTSPC(32)",
{GRN}+*{HOME}{BLU}"
:rem 27
1100 GOSUB1600:IFFL=1THEN1220
:rem 198
1110 IFTEMP=1THENTEMP=0:GOTO1210:
:rem 255
1120 IFOUT<3THEN1210
:rem 173
1130 IFNP=1THEN1660
:rem 92
1140 IFIN>=3ANDCK=1ANDS(1)<>S(2)THEN1660
:rem 93
1150 OUT=0:TEMP=1:POKEFB,32:POKESB,32:POK
ETB,32
:rem 171
1160 PRINT"{HOME}{ 14 GIU'}{RVS}{CYN}
{ 4 SPAZI}CAMBIO BATTITORI":FORI=1TO
1000:NEXT
:rem 93
1170 PRINT"{SU}"SPC(4)"{RVS}{ 16 SPAZI}":
IFNP=1THENCK=0:IN=IN+1:PL=1:GOTO1190
:rem 110
1180 CK=CK+1:IFCK=2THENCK=0:IN=IN+1
:rem 112
1190 IFPL=1THENPL=2:P1$="{RVS}HOME":P2$="
VISITOR":GOTO1100
:rem 255
1200 PL=1:P1$="HOME":P2$="{RVS}VISITOR":G
OTO1100
:rem 46
1210 IFPEEK(53250)>90THENFORA=PEEK(53250)
TO90STEP-1:POKE53250,A:NEXT:
:rem 108
1220 SYS49152:IFPEEK(930)THENPOKE54283,16
:POKE54283,17
:rem 158
1230 IFFL=1THENRM=RM-1:PRINT"{HOME}

```



```

    { 18 GIU' } { GRN } "SPC(28)" { RVS } PALLE
    "RM" { SIN } { RVS } " :rem 194
1240 IF FL=1 AND RM=0 THEN 1690 :rem 52
1250 IF FL=1 THEN 1210 :rem 74
1260 PRINT " { HOME } " : KT=54272 :rem 199
1270 MT=PEEK(930) : IF MT>4 THEN OUT=OUT+1 : SK=
    0 : GOTO 1100 :rem 200
1280 IF MT=4 THEN GOSUB 1580 :rem 233
1290 IF MT=0 THEN SK=SK+1 : IF SK=3 THEN OUT=OUT+
    1 : SK=0 : GOTO 1100 :rem 118
1300 IF MT=0 THEN 1100 :rem 82
1310 TM=MT : SK=0 :rem 137
1320 TE=160 : ET=160 : G0=CL(PL) :rem 68
1330 A=2017 : FB=1822 : SB=1617 : TB=1812 : P1=PE
    EK(FB) : P2=PEEK(SB) : P3=PEEK(TB)
    :rem 143
1340 GOSUB 1510 : GOSUB 1530 : B1=205 : B2=206 : B3
    =205 : M0=A-39 : M1=FB-41 : M2=SB+39
    :rem 24
1350 M3=TB+41 : G1=5 : G2=5 : G3=5 :rem 27
1360 IF P1=160 THEN B1=160 : G1=CL(PL)
    :rem 143
1370 IF P2=160 THEN B2=160 : G2=CL(PL)
    :rem 147
1380 IF P3=160 THEN B3=160 : G3=CL(PL)
    :rem 151
1390 X=CL(PL) : FOR I=1 TO 4 : GOSUB 1530 : GOSUB 15
    70 :rem 151
1400 IF MT=4 THEN GOSUB 1640 :rem 224
1410 POKEM0, TE : POKEM0+KT, G0 : M0=M0-39 : POKE
    M1, B1 : POKEM1+KT, G1 :rem 111
1420 M1=M1-41 : POKEM2, B2 : POKEM2+KT, G2 : M2=M
    2+39 : POKEM3, B3 : POKEM3+KT, G3 : M3=M3+41
    :rem 103
1430 NEXT : GOSUB 1530 : POKEFB, ET : POKEFB+KT, G
    0 :rem 158
1440 IF B1=160 THEN POKESB, 160 : POKESB+54272,
    G1 :rem 38
1450 IF B2=160 THEN POKETB, 160 : POKETB+54272,
    G2 :rem 43
1460 IF B3=160 THEN POKEA, 160 : POKEA+54272, G3
    :rem 132
1470 IF MT>1 THEN TE=206 : ET=32 :rem 125
1480 IF PEEK(A)=160 THEN S(PL)=S(PL)+1 : POKEA
    , 43 : POKEA+KT, 5 : GOSUB 1590 : GOSUB 1600
    :rem 75
1490 TM=TM-1 : IF TM=0 THEN 1100 :rem 115
1500 G0=5 : GOTO 1330 :rem 233
1510 PRINT " { HOME } { 14 GIU' } "SPC(33)" "
    :rem 122
1520 PRINT " { 4 GIU' } "SPC(28)" "SPC(9)" " :
    PRINT " { 4 GIU' } "SPC(33)" { GRN } + { HOME }
    " : RETURN :rem 28
1530 PRINT " { HOME } { 15 GIU' } "SPC(32);
    :rem 131
1540 PRINT " { RVS } { GRN } { 1 SH N } { OFF } 3 { RVS }
    { 1 SH M } { GIU' } { 4 SIN } { 1 SH N }
    { OFF } * { RVS } { OFF }, { RVS } { 1 SH M }
    { GIU' } { 6 SIN } { 1 SH N } { OFF } * { RVS }
    { 3 SPAZI } { OFF }, { RVS } { 1 SH M } { GIU' }
    { 8 SIN } { 1 SH N } { OFF } * { RVS }
    { 5 SPAZI } { OFF }, { RVS } { 1 SH M } "
    :rem 37

```

```

1550 PRINT " { GIU' } "SPC(29)" { RVS } { 1 SH M }
    { OFF } . { RVS } { 5 SPAZI } { OFF } - { RVS }
    { 1 SH N } { GIU' } { 8 SIN } { 1 SH M }
    { OFF } . { RVS } { 3 SPAZI } { OFF } - { RVS }
    { 1 SH N } " :rem 26
1560 PRINTSPC(31)" { RVS } { 1 SH M } { OFF } .
    { RVS } { OFF } - { RVS } { 1 SH N } { GIU' }
    { 4 SIN } { 1 SH M } { OFF } 4 { RVS } { 1 SH N }
    { HOME } " : RETURN :rem 173
1570 POKE54273, 40 : POKE54277, 1 : POKE54296, 1
    5 : POKE54276, 128 : POKE54276, 129 : RETURN
    :rem 240
1580 POKE54287, 40 : POKE54291, 220 : POKE54290
    , 128 : POKE54290, 129 : RETURN :rem 72
1590 POKE54273, 70 : POKE54277, 41 : POKE54296,
    15 : POKE54276, 16 : POKE54276, 17 : RETURN
    :rem 193
1600 IF FL=1 THEN PRINT " { HOME } { 18 GIU' }
    { GRN } "SPC(28)" { RVS } PITCHES "RM : RETURN
    :rem 121
1610 PRINT " { HOME } { GIU' } { BLU } "SPC(28) P2$ "
    { RVS } "S(2) : PRINTSPC(28)" { GIU' } "P1$ "
    { RVS } { 3 SPAZI } "S(1) :rem 220
1620 PRINTSPC(28)" { 2 GIU' } { RVS } INNING "I
    N : PRINTSPC(28)" { GIU' } { RVS } STRIKE "SK
    :rem 8
1630 PRINTSPC(28)" { GIU' } { RVS } OUT
    { 4 SPAZI } "OUT : RETURN :rem 105
1640 POKE53281, X : IF X=1 THEN X=CL(PL) : RETURN
    :rem 106
1650 X=1 : RETURN :rem 172
1660 PRINT " { HOME } "SPC(5)" { RVS } { CYN } G A M
    E { 2 SPAZI } O V E R { 13 GIU' } "
    :rem 69
1670 PRINT " { RVS } { 3 SPAZI } UN TASTO PER GI
    OCARE " : POKE198, 0 : WAIT198, 1 : GETMT$ : C
    LR :rem 143
1680 POKE53280, 0 : POKE53281, 0 : PRINT " { CLR } "
    : GOTO 730 :rem 207
1690 PRINT " { HOME } { RVS } { CYN } G A M E
    { 2 SPAZI } O V E R { 13 GIU' } " : GOTO 167

```


High rise

di C. McGuyer
trad. e adatt. di L. Priotto

State lavorando alla costruzione di un grattacielo e vi trovate intrappolati nelle sue strutture all'ultimo piano mentre sta scendendo la sera: riuscirete a scendere indenni fino al piano terra, evitando gli attacchi di un volatile gigante e del robot di guardia alla costruzione? E' richiesto l'utilizzo di un joystick in Porta 2.

Avete appena terminato di imbullonare l'ultima putrella della giornata quando vi guardate intorno e vi accorgete di essere rimasti soli: all'ultimo piano dell'altissimo grattacielo in costruzione si ode solo il sibilo del vento nella luce ormai scarsa del sole al tramonto.

Vi accingete a scendere quando un pensiero vi gela il sangue: proprio ieri è stato installato nella costruzione un robot di guardia, con il compito di distruggere ogni eventuale intruso. Oltre a ciò, la costruzione è popolata da combattivi uccellacci che hanno nidificato nei locali vuoti.

Dovrete affidarvi alla velocità e prontezza di riflessi per riuscire a raggiungere sani e salvi l'uscita molti piani sotto di voi...

Potete usare gli ascensori, che di notte si muovono a caso tra un piano e l'altro, fermandosi altrettanto a caso a un piano qualunque. Se un ascensore è fermo al piano in cui vi trovate, vi basterà salirvi sopra per fare in modo che esso si muova: non potete tuttavia sapere a

priori se l'ascensore si sposterà verso l'alto oppure verso il basso. Il robot anti-ladro cercherà di fare in modo che cadiate all'esterno dell'edificio e se riesce a scorgervi a bordo di un ascensore, emetterà una scarica elettrica, che provocherà inesorabilmente la vostra fine. Dovrete inoltre evitare il pericolo che uno degli uccellacci vi artigli e vi trasporti via in volo, anche mentre siete all'interno di un ascensore...

Gli uccellacci e il robot presentano oltre a tutto un pericolo combinato: quando

uno dei primi viene colpito dal robot, precipita a peso morto fino a terra, con il rischio di travolgervi, se vi trovate lungo la sua traiettoria.

Iniziate il gioco con cinque vite a disposizione, al decimo piano della costruzione. Ogni piano guadagnato verso il basso vi frutterà 100 punti e se riuscite a uscire sani e salvi dalla costruzione, potrete riprendere il gioco partendo da un piano più alto.

Note per il caricamento, la dupli-

cazione su disco e la digitazione del programma

Se si dispone della cassetta allegata alla rivista, il programma si carica in memoria e si manda in esecuzione nel modo seguente:

1. Posizionare il nastro in corrispondenza dell'inizio del programma.

2. Spegner e riaccendere il computer e digitare (senza inserire spazi tra le virgolette):

```
LOAD"",1,1 (RETURN)
```

3. A caricamento avvenuto (ricomparsa del cursore con la scritta READY) attivare il programma digitando:

```
NEW (RETURN)  
SYS 49152 (RETURN)
```

La cassetta contiene, nello stesso ordine in cui appaiono sulla rivista, tutti i programmi di questo numero: per ottenerne un elenco, con la relativa posizione del

contanastro, inserire nel registratore la cassetta completamente riavvolta, azzerare il contanastro e digitare:

VERIFY ""

Poichè la cassetta sicuramente non contiene alcun programma dal nome "", verranno passati in rassegna tutti i programmi contenuti attraverso il messaggio:

FOUND nome del programma

Risulta utile annotarsi i nomi dei vari programmi e la relativa posizione del contanastro (sottrarre due o tre unità dal valore corrispondente all'apparizione del messaggio FOUND), in modo da poter in seguito caricare il programma desiderato senza dover necessariamente leggere tutti i programmi precedenti.

Il programma, a causa dell'area di memoria occupata, non può essere trasferito su disco con una semplice istruzione SAVE. Per effettuare il trasferimento su disco occorrerà invece caricare e attivare MLX (in altra parte della rivista), ri-

spondere alle sue domande con gli indirizzi riportati più sotto, premere quindi contemporaneamente i tasti SHIFT e L per caricare il programma da cassetta, e a caricamento avvenuto premere contemporaneamente i tasti SHIFT e S per salvarlo su disco.

Gli abbonati che non dispongono della cassetta dovranno invece digitare il programma attraverso MLX. Una volta caricato e attivato MLX si dovrà rispondere alle sue due domande iniziali con i seguenti valori:

indirizzo iniziale: 49152

indirizzo finale : 52863

Una volta salvato il programma su nastro esso potrà essere ricaricato in memoria in qualsiasi momento, senza più bisogno di MLX, seguendo le istruzioni date più sopra per i possessori di cassetta.

Si ricorda che se il programma è stato salvato su disco, occorre aggiungere il nome del file e il numero di periferica 8 all'istruzione LOAD, mantenendo tuttavia l'indirizzo secondario 1.

High Rise

49152	:169,000,168,133,251,169,122	49338	:006,157,192,007,232,224,236
49158	:032,133,252,169,000,133,213	49344	:031,208,234,172,176,201,190
49164	:253,169,203,133,254,177,177	49350	:192,010,240,045,185,166,012
49170	:253,145,251,200,208,249,044	49356	:201,133,251,200,185,166,060
49176	:230,254,230,252,165,252,127	49362	:201,133,252,200,140,176,032
49182	:201,036,208,239,160,024,130	49368	:201,162,004,160,000,169,144
49188	:185,238,201,153,000,212,001	49374	:027,145,251,202,240,016,079
49194	:136,016,247,160,004,169,006	49380	:024,165,251,105,040,133,178
49200	:048,153,136,202,136,016,227	49386	:251,165,252,105,000,133,116
49206	:250,032,231,200,032,068,099	49392	:252,076,221,192,076,195,228
49212	:229,169,005,141,081,202,119	49398	:192,160,000,140,176,201,091
49218	:169,049,141,095,202,169,123	49404	:169,128,162,000,157,248,092
49224	:048,141,096,202,169,000,216	49410	:007,232,224,005,208,248,158
49230	:141,032,208,141,033,208,073	49416	:169,255,141,021,208,162,196
49236	:141,235,201,141,136,201,115	49422	:000,173,134,201,157,001,168
49242	:160,004,185,136,202,153,162	49428	:208,232,232,024,173,134,255
49248	:145,007,136,016,247,032,167	49434	:201,105,032,141,134,201,072
49254	:167,199,032,222,198,173,069	49440	:224,010,240,003,076,015,088
49260	:151,201,133,251,032,139,247	49446	:193,169,050,141,134,201,158
49266	:192,024,173,151,201,105,192	49452	:169,255,141,028,208,169,246
49272	:007,141,151,201,201,036,089	49458	:015,141,037,208,169,002,110
49278	:240,003,076,107,192,169,145	49464	:141,038,208,162,000,173,010
49284	:001,141,151,201,076,171,105	49470	:135,201,157,000,208,232,227
49290	:192,169,004,133,252,162,026	49476	:232,024,173,135,201,105,170
49296	:025,160,000,169,028,145,159	49482	:056,141,135,201,224,010,073
49302	:251,202,240,016,024,165,024	49488	:240,003,076,061,193,169,054
49308	:251,105,040,133,251,165,077	49494	:024,141,135,201,160,001,236
49314	:252,105,000,133,252,076,212	49500	:185,095,202,153,162,006,127
49320	:147,192,096,162,000,169,166	49506	:136,016,247,169,000,141,039
49326	:029,157,160,004,157,104,017	49512	:235,201,169,240,141,010,076
49332	:005,157,048,006,157,248,033	49518	:208,169,184,141,011,208,007
		49524	:169,124,141,012,208,173,175

49530 :030,208,169,104,141,013,019
 49536 :208,169,020,141,014,208,120
 49542 :141,015,208,169,138,141,178
 49548 :255,007,076,216,193,138,001
 49554 :072,162,004,160,000,136,168
 49560 :208,253,202,208,250,104,097
 49566 :170,096,172,152,201,185,110
 49572 :146,201,221,001,208,240,157
 49578 :009,200,192,005,240,013,061
 49584 :140,152,201,096,169,002,168
 49590 :157,136,201,032,097,194,231
 49596 :096,160,000,140,152,201,169
 49602 :096,165,197,201,063,240,132
 49608 :005,201,004,240,007,096,241
 49614 :169,000,141,024,212,000,240
 49620 :076,183,195,096,162,000,156
 49626 :032,145,193,032,201,195,248
 49632 :032,105,196,032,059,195,075
 49638 :032,160,193,032,107,194,180
 49644 :032,195,193,032,027,197,144
 49650 :032,178,196,032,189,194,039
 49656 :032,047,200,189,136,201,029
 49662 :201,001,240,065,144,002,139
 49668 :176,016,254,001,208,189,080
 49674 :001,208,201,224,240,037,153
 49680 :032,097,194,076,218,193,058
 49686 :222,192,201,189,192,201,195
 49692 :240,003,076,218,193,189,179
 49698 :137,201,157,136,201,032,130
 49704 :097,194,032,154,200,041,246
 49710 :031,157,192,201,076,218,153
 49716 :193,169,001,157,136,201,141
 49722 :157,137,201,032,097,194,108
 49728 :076,218,193,222,001,208,214
 49734 :189,001,208,201,064,240,205
 49740 :006,032,097,194,076,218,187
 49746 :193,169,000,157,136,201,170
 49752 :157,137,201,032,097,194,138
 49758 :076,218,193,232,232,224,245
 49764 :010,240,001,096,162,000,097
 49770 :096,160,005,185,146,201,131
 49776 :205,011,208,240,004,136,148
 49782 :016,245,096,173,000,220,100
 49788 :041,015,201,011,240,029,149
 49794 :201,007,240,011,169,000,246
 49800 :141,011,212,169,133,141,175
 49806 :253,007,096,173,010,208,121
 49812 :201,245,240,248,032,244,078
 49818 :194,032,173,194,096,173,248
 49824 :010,208,201,023,240,234,052
 49830 :032,017,195,032,173,194,041
 49836 :096,169,129,141,011,212,162
 49842 :169,034,141,008,212,169,143
 49848 :128,141,011,212,096,206,210
 49854 :084,202,173,084,202,240,151
 49860 :001,096,172,082,202,185,166
 49866 :011,202,141,000,212,185,185
 49872 :012,202,141,001,212,169,177
 49878 :017,141,004,212,200,200,220

49884 :192,032,240,009,140,082,147
 49890 :202,169,035,141,084,202,035
 49896 :096,160,000,140,082,202,144
 49902 :169,035,141,084,202,096,197
 49908 :238,010,208,173,234,201,028
 49914 :201,020,240,005,201,040,189
 49920 :240,006,096,169,136,076,211
 49926 :013,195,169,137,076,013,097
 49932 :195,141,253,007,096,206,142
 49938 :010,208,173,234,201,201,021
 49944 :020,240,005,201,040,240,002
 49950 :006,096,169,134,076,013,012
 49956 :195,169,135,076,013,195,051
 49962 :032,145,193,206,101,202,153
 49968 :173,101,202,208,245,169,122
 49974 :000,141,101,202,096,173,255
 49980 :030,208,141,030,208,096,005
 49986 :032,145,193,238,011,208,125
 49992 :032,034,201,173,011,208,219
 49998 :201,245,208,240,169,000,117
 50004 :141,004,212,169,176,141,159
 50010 :110,202,172,081,202,169,002
 50016 :032,153,216,005,136,240,110
 50022 :040,140,081,202,032,042,127
 50028 :195,173,030,208,169,184,043
 50034 :141,011,208,169,064,141,080
 50040 :013,208,141,012,208,169,103
 50046 :020,141,015,208,169,000,167
 50052 :141,235,201,141,100,202,128
 50058 :104,104,076,216,193,024,087
 50064 :162,011,160,005,032,240,242
 50070 :255,169,255,141,027,208,181
 50076 :160,000,185,141,202,240,060
 50082 :007,032,210,255,200,076,174
 50088 :158,195,032,183,195,169,076
 50094 :000,141,027,208,104,104,246
 50100 :076,058,192,169,000,141,048
 50106 :024,212,173,000,220,041,088
 50112 :016,208,244,169,127,141,073
 50118 :024,212,096,173,030,208,173
 50124 :141,158,201,208,004,096,244
 50130 :076,102,196,160,004,173,153
 50136 :158,201,217,161,201,240,114
 50142 :073,201,096,240,239,136,183
 50148 :016,241,160,000,173,010,060
 50154 :208,217,000,208,144,037,024
 50160 :056,249,000,208,201,004,190
 50166 :176,022,173,011,208,217,029
 50172 :001,208,144,031,056,249,173
 50178 :001,208,201,010,176,006,092
 50184 :185,001,208,141,011,208,250
 50190 :200,200,192,010,208,212,012
 50196 :096,056,185,000,208,237,034
 50202 :010,208,076,244,195,185,176
 50208 :001,208,237,011,208,076,005
 50214 :004,196,138,072,162,050,148
 50220 :169,129,153,248,007,140,122
 50226 :236,201,032,154,200,141,246
 50232 :008,212,169,129,141,011,214

50238 :212,032,145,193,172,236,028
 50244 :201,169,130,153,248,007,208
 50250 :140,236,201,032,154,200,013
 50256 :141,008,212,032,145,193,043
 50262 :172,236,201,202,016,208,097
 50268 :169,128,153,248,007,141,170
 50274 :011,212,104,170,076,066,225
 50280 :195,173,237,201,206,237,073
 50286 :201,240,001,096,169,003,052
 50292 :141,237,201,032,213,199,115
 50298 :032,036,199,160,005,185,227
 50304 :146,201,205,013,208,240,117
 50310 :004,136,016,245,096,173,036
 50316 :159,201,208,017,173,012,142
 50322 :208,201,248,240,004,238,005
 50328 :012,208,096,169,001,141,011
 50334 :159,201,096,173,012,208,239
 50340 :201,024,240,004,206,012,083
 50346 :208,096,169,000,141,159,175
 50352 :201,096,206,234,201,173,007
 50358 :234,201,240,005,201,020,059
 50364 :240,069,096,169,131,141,010
 50370 :254,007,169,139,141,255,135
 50376 :007,169,041,141,234,201,225
 50382 :173,015,208,201,050,144,229
 50388 :040,201,120,176,036,172,189
 50394 :080,202,185,007,202,141,011
 50400 :014,212,185,008,202,141,218
 50406 :015,212,169,017,141,018,034
 50412 :212,200,200,192,004,240,004
 50418 :004,140,080,202,096,160,156
 50424 :000,140,080,202,096,169,167
 50430 :032,141,018,212,096,169,154
 50436 :132,141,254,007,169,138,077
 50442 :141,255,007,096,169,064,230
 50448 :141,013,208,169,000,141,176
 50454 :235,201,076,080,197,173,216
 50460 :011,208,201,224,240,001,145
 50466 :096,138,072,032,058,201,119
 50472 :141,001,212,032,145,193,252
 50478 :056,173,011,208,233,008,223
 50484 :141,011,208,173,015,208,040
 50490 :201,050,144,005,233,008,187
 50496 :141,015,208,173,013,208,054
 50502 :201,072,144,196,056,233,204
 50508 :008,141,013,208,032,222,188
 50514 :198,032,017,199,160,000,176
 50520 :169,193,133,251,169,007,242
 50526 :133,252,185,075,202,141,058
 50532 :001,212,169,017,141,004,132
 50538 :212,169,028,145,251,024,167
 50544 :165,251,105,007,133,251,000
 50550 :201,228,240,003,076,107,205
 50556 :197,238,097,202,173,160,167
 50562 :201,206,160,201,016,157,047
 50568 :169,004,141,160,201,173,216
 50574 :030,208,162,000,169,029,228
 50580 :157,192,007,232,224,031,223
 50586 :208,246,169,004,141,160,058

50592 :201,173,030,208,169,000,173
 50598 :141,097,202,162,001,222,223
 50604 :162,006,189,162,006,201,130
 50610 :049,240,082,201,047,240,013
 50616 :059,162,002,254,017,005,171
 50622 :189,017,005,201,058,240,132
 50628 :077,162,000,189,017,005,134
 50634 :221,145,007,240,024,144,215
 50640 :014,189,017,005,157,145,223
 50646 :007,157,136,202,232,224,148
 50652 :005,208,242,173,083,202,109
 50658 :208,063,104,170,096,232,075
 50664 :224,005,208,219,173,083,120
 50670 :202,208,050,104,170,096,044
 50676 :169,057,157,162,006,202,229
 50682 :189,162,006,201,048,240,072
 50688 :184,222,162,006,076,185,067
 50694 :197,202,189,162,006,201,195
 50700 :048,240,012,076,185,197,002
 50706 :169,048,157,017,005,202,104
 50712 :076,187,197,169,001,141,027
 50718 :083,202,076,185,197,169,174
 50724 :000,141,083,202,024,173,147
 50730 :096,202,105,005,141,096,175
 50736 :202,201,058,240,003,076,060
 50742 :064,198,169,048,141,096,002
 50748 :202,238,095,202,024,160,213
 50754 :004,162,011,032,240,255,002
 50760 :160,000,185,195,202,240,030
 50766 :007,032,210,255,200,076,090
 50772 :074,198,169,224,141,011,133
 50778 :208,141,010,208,169,134,192
 50784 :141,253,007,206,010,208,153
 50790 :032,145,193,032,145,193,074
 50796 :169,135,141,253,007,206,251
 50802 :010,208,169,000,141,004,134
 50808 :212,141,018,212,032,173,140
 50814 :194,032,145,193,032,145,099
 50820 :193,173,010,208,201,002,151
 50826 :208,210,032,198,198,024,240
 50832 :160,004,162,011,032,240,241
 50838 :255,160,000,185,220,202,148
 50844 :240,007,032,210,255,200,076
 50850 :076,153,198,032,042,195,090
 50856 :160,024,169,032,153,188,126
 50862 :005,136,016,250,169,028,010
 50868 :141,192,005,141,199,005,095
 50874 :141,206,005,032,222,198,222
 50880 :104,104,104,076,090,193,095
 50886 :160,000,185,217,005,201,198
 50892 :032,240,006,200,192,005,111
 50898 :208,244,096,169,030,153,086
 50904 :217,005,238,081,202,096,031
 50910 :169,000,133,251,169,216,136
 50916 :133,252,138,072,162,025,242
 50922 :160,000,169,002,145,251,193
 50928 :200,192,031,208,249,169,009
 50934 :010,145,251,200,192,039,059
 50940 :208,249,024,165,251,105,230

0946 : 040, 133, 251, 165, 252, 105, 180
 0952 : 000, 133, 252, 202, 208, 220, 255
 0958 : 104, 170, 096, 173, 019, 007, 071
 0964 : 201, 027, 240, 006, 169, 027, 178
 0970 : 141, 219, 007, 096, 169, 027, 173
 0976 : 141, 195, 007, 096, 160, 005, 124
 0982 : 185, 177, 201, 205, 012, 208, 002
 0988 : 240, 007, 136, 016, 245, 173, 093
 0994 : 031, 208, 096, 173, 191, 201, 182
 1000 : 208, 056, 173, 013, 208, 201, 147
 1006 : 224, 240, 082, 173, 235, 201, 193
 1012 : 208, 020, 160, 003, 173, 031, 151
 1018 : 208, 217, 183, 201, 240, 004, 103
 1024 : 136, 016, 248, 096, 169, 001, 234
 1030 : 141, 235, 201, 096, 238, 013, 242
 1036 : 208, 032, 156, 199, 096, 169, 184
 1042 : 000, 141, 235, 201, 173, 031, 111
 1048 : 208, 173, 159, 201, 073, 001, 151
 1054 : 141, 159, 201, 096, 173, 013, 125
 1060 : 208, 201, 064, 240, 026, 173, 004
 1066 : 235, 201, 208, 014, 160, 003, 175
 1072 : 173, 031, 208, 217, 187, 201, 121
 1078 : 240, 204, 136, 016, 248, 096, 050
 1084 : 206, 013, 208, 032, 156, 199, 186
 1090 : 096, 173, 191, 201, 073, 001, 113
 1096 : 141, 191, 201, 096, 238, 235, 230
 1102 : 201, 173, 235, 201, 201, 041, 186
 1108 : 240, 187, 096, 160, 004, 185, 012
 1114 : 111, 202, 153, 193, 004, 185, 250
 1120 : 116, 202, 153, 137, 005, 185, 206
 1126 : 121, 202, 153, 081, 006, 185, 162
 1132 : 126, 202, 153, 025, 007, 185, 118
 1138 : 131, 202, 153, 065, 007, 169, 153
 1144 : 030, 153, 217, 005, 169, 048, 054
 1150 : 153, 017, 005, 136, 016, 213, 234
 1156 : 096, 173, 100, 202, 208, 021, 244
 1162 : 173, 015, 208, 205, 011, 208, 014
 1168 : 176, 055, 173, 098, 202, 208, 112
 1174 : 014, 032, 154, 200, 201, 191, 254
 1180 : 240, 007, 096, 169, 141, 141, 006
 1186 : 255, 007, 096, 169, 001, 141, 143
 1192 : 098, 202, 173, 015, 208, 205, 125
 1198 : 011, 208, 240, 027, 238, 015, 225
 1204 : 208, 173, 014, 208, 205, 010, 054
 1210 : 208, 240, 016, 176, 006, 238, 126
 1216 : 014, 208, 238, 014, 208, 206, 136
 1222 : 014, 208, 096, 201, 240, 240, 253
 1228 : 007, 206, 014, 208, 238, 015, 204
 1234 : 208, 096, 169, 000, 141, 098, 234
 1240 : 202, 169, 032, 141, 015, 208, 039
 1246 : 096, 173, 100, 202, 208, 053, 110
 1252 : 173, 015, 208, 205, 011, 208, 104
 1258 : 176, 027, 173, 010, 208, 205, 089
 1264 : 014, 208, 144, 027, 056, 237, 238
 1270 : 014, 208, 201, 008, 176, 011, 176
 1276 : 056, 173, 011, 208, 237, 015, 008
 1282 : 208, 201, 007, 144, 077, 173, 124
 1288 : 030, 208, 201, 192, 240, 011, 202
 1294 : 096, 056, 173, 014, 208, 237, 110

51300 : 010, 208, 076, 072, 200, 169, 067
 51306 : 001, 141, 100, 202, 238, 015, 035
 51312 : 208, 032, 154, 200, 141, 015, 094
 51318 : 212, 169, 021, 141, 018, 212, 123
 51324 : 173, 030, 208, 201, 160, 240, 112
 51330 : 020, 173, 015, 208, 201, 245, 224
 51336 : 240, 001, 096, 169, 000, 141, 015
 51342 : 015, 208, 141, 100, 202, 173, 213
 51348 : 030, 208, 096, 076, 066, 195, 051
 51354 : 138, 072, 032, 158, 224, 104, 114
 51360 : 170, 165, 098, 096, 169, 016, 106
 51366 : 141, 004, 212, 169, 138, 141, 203
 51372 : 255, 007, 169, 136, 141, 253, 109
 51378 : 007, 032, 145, 193, 173, 015, 231
 51384 : 208, 141, 011, 208, 173, 014, 171
 51390 : 208, 141, 010, 208, 206, 015, 210
 51396 : 208, 169, 139, 141, 255, 007, 091
 51402 : 032, 145, 193, 032, 154, 200, 190
 51408 : 141, 015, 212, 169, 017, 141, 135
 51414 : 018, 212, 173, 015, 208, 201, 017
 51420 : 025, 208, 197, 169, 016, 141, 208
 51426 : 018, 212, 076, 092, 195, 120, 171
 51432 : 169, 051, 133, 001, 169, 000, 243
 51438 : 133, 251, 133, 253, 169, 208, 105
 51444 : 133, 252, 169, 048, 133, 254, 209
 51450 : 162, 002, 160, 000, 177, 251, 234
 51456 : 145, 253, 136, 208, 249, 230, 197
 51462 : 252, 230, 254, 202, 208, 240, 112
 51468 : 169, 055, 133, 001, 088, 169, 115
 51474 : 028, 141, 024, 208, 160, 031, 098
 51480 : 185, 202, 201, 153, 216, 048, 005
 51486 : 136, 016, 247, 096, 172, 110, 039
 51492 : 202, 140, 001, 212, 169, 017, 009
 51498 : 141, 004, 212, 136, 140, 110, 017
 51504 : 202, 169, 000, 141, 011, 212, 015
 51510 : 141, 018, 212, 096, 169, 004, 182
 51516 : 133, 252, 169, 000, 133, 251, 230
 51522 : 162, 030, 024, 165, 251, 105, 035
 51528 : 040, 133, 253, 165, 252, 105, 252
 51534 : 000, 133, 254, 160, 000, 177, 034
 51540 : 253, 145, 251, 230, 251, 165, 099
 51546 : 251, 208, 002, 230, 252, 202, 211
 51552 : 016, 226, 024, 165, 251, 105, 115
 51558 : 009, 133, 251, 165, 252, 105, 249
 51564 : 000, 133, 252, 165, 251, 201, 086
 51570 : 192, 208, 205, 165, 252, 201, 057
 51576 : 007, 208, 199, 169, 032, 160, 127
 51582 : 030, 153, 192, 007, 136, 016, 148
 51588 : 250, 096, 064, 024, 000, 000, 054
 51594 : 000, 000, 000, 000, 000, 000, 138
 51600 : 000, 000, 064, 104, 144, 184, 128
 51606 : 224, 001, 000, 033, 034, 036, 222
 51612 : 040, 048, 000, 000, 004, 097, 089
 51618 : 098, 100, 104, 112, 027, 004, 095
 51624 : 203, 004, 171, 005, 091, 006, 136
 51630 : 059, 007, 000, 039, 040, 041, 104
 51636 : 231, 232, 233, 191, 159, 063, 009
 51642 : 031, 255, 223, 127, 095, 000, 149
 51648 : 004, 004, 004, 004, 004, 004, 216

51654 :004,004,004,004,130,084,172
 51660 :130,084,130,084,130,084,078
 51666 :032,016,032,016,032,016,098
 51672 :032,016,000,000,000,000,008
 51678 :255,195,195,255,008,028,134
 51684 :008,028,042,093,020,020,183
 51690 :021,000,000,020,000,000,019
 51696 :000,000,000,069,240,000,037
 51702 :000,000,000,000,069,240,043
 51708 :000,000,000,000,017,088,101
 51714 :088,000,000,002,127,030,249
 51720 :025,031,021,071,005,071,232
 51726 :005,237,005,071,005,071,152
 51732 :006,071,005,012,007,071,192
 51738 :006,071,005,071,005,237,165
 51744 :005,071,005,071,006,071,005
 51750 :005,012,007,071,006,031,170
 51756 :021,031,021,031,021,031,200
 51762 :021,181,023,181,023,181,148
 51768 :023,181,023,030,025,030,112
 51774 :025,030,025,030,025,181,122
 51780 :023,181,023,181,023,181,168
 51786 :023,049,223,135,162,000,154
 51792 :000,005,000,000,025,000,110
 51798 :247,220,192,164,136,108,129
 51804 :080,052,024,049,048,000,089
 51810 :000,000,000,000,013,007,118
 51816 :006,005,014,011,010,009,159
 51822 :161,019,003,015,018,005,075
 51828 :013,005,014,032,032,006,218
 51834 :012,015,015,018,008,009,199
 51840 :007,008,032,019,003,015,212
 51846 :018,005,048,048,048,048,093
 51852 :048,030,089,079,085,039,254
 51858 :086,069,032,082,085,078,066
 51864 :032,079,085,084,032,079,031
 51870 :070,032,077,069,078,013,241
 51876 :013,029,029,029,029,029,066
 51882 :080,082,069,083,083,032,087
 51888 :070,073,082,069,032,084,074
 51894 :079,032,080,076,065,089,091
 51900 :032,065,071,065,073,078,060
 51906 :000,071,082,079,085,078,077
 51912 :068,032,070,076,079,079,092
 51918 :082,044,089,079,085,039,112
 51924 :082,069,032,083,065,070,101
 51930 :069,000,078,079,087,032,051
 51936 :087,069,032,071,079,032,082
 51942 :065,032,076,073,084,084,132
 51948 :076,069,032,072,073,071,117
 51954 :072,069,082,000,254,000,207
 51960 :236,000,238,000,110,098,162
 51966 :002,098,000,000,000,000,098
 51972 :000,000,000,000,000,000,004
 51978 :000,000,010,170,128,008,070
 51984 :000,128,008,000,128,008,032
 51990 :000,128,008,000,128,008,038
 51996 :000,128,008,000,128,008,044
 52002 :000,128,008,000,128,008,050

52008 :000,128,008,000,128,008,056
 52014 :000,128,008,000,128,008,062
 52020 :000,128,008,000,128,008,068
 52026 :000,128,010,170,128,131,113
 52032 :064,016,000,068,001,001,214
 52038 :016,004,004,004,001,004,103
 52044 :010,170,144,008,000,128,024
 52050 :072,000,132,024,000,128,182
 52056 :072,000,129,024,000,132,189
 52062 :072,000,144,072,000,132,002
 52068 :008,000,128,008,000,129,117
 52074 :024,000,132,072,000,129,207
 52080 :024,000,132,072,000,144,228
 52086 :072,000,132,008,000,129,203
 52092 :010,170,132,131,064,000,119
 52098 :016,016,068,064,004,016,058
 52104 :017,016,000,004,010,170,097
 52110 :144,072,000,132,024,000,002
 52116 :129,072,000,132,024,000,249
 52122 :144,008,000,132,008,000,190
 52128 :128,024,000,128,072,000,000
 52134 :144,024,000,132,008,000,218
 52140 :129,024,000,132,072,000,017
 52146 :132,072,000,129,008,000,007
 52152 :132,072,000,144,074,170,008
 52158 :128,131,000,000,000,000,193
 52164 :000,000,000,000,000,000,196
 52170 :000,000,000,000,000,017,219
 52176 :017,016,000,000,000,017,002
 52182 :017,016,000,168,000,000,159
 52188 :136,000,000,016,000,000,116
 52194 :016,000,051,051,048,000,136
 52200 :032,000,051,051,048,000,158
 52206 :032,000,000,032,000,000,046
 52212 :168,000,000,136,000,002,038
 52218 :138,000,002,170,000,255,047
 52224 :000,000,000,000,000,000,000
 52230 :000,000,000,000,000,000,006
 52236 :000,000,000,004,068,064,148
 52242 :016,000,016,004,068,064,186
 52248 :000,168,000,000,136,000,072
 52254 :000,016,000,000,016,000,062
 52260 :012,204,192,048,032,048,060
 52266 :012,204,192,000,032,000,226
 52272 :000,032,000,000,168,000,248
 52278 :000,136,000,002,138,000,074
 52284 :002,170,000,255,000,000,231
 52290 :000,000,000,000,000,000,066
 52296 :000,000,000,000,000,000,072
 52302 :000,000,012,000,000,051,141
 52308 :000,000,004,000,000,004,092
 52314 :000,000,042,000,000,162,038
 52320 :128,000,170,128,002,034,046
 52326 :032,002,042,032,003,034,247
 52332 :048,000,021,000,000,017,194
 52338 :000,000,017,000,000,017,148
 52344 :000,000,017,000,000,255,136
 52350 :192,255,000,000,000,000,061
 52356 :000,000,000,000,000,000,132

52362 :000,000,000,000,000,000,138
 52368 :012,000,000,051,000,000,207
 52374 :004,000,000,004,000,000,158
 52380 :010,000,003,010,128,000,051
 52386 :138,032,000,043,128,000,247
 52392 :005,000,000,021,000,000,194
 52398 :068,000,000,068,000,000,054
 52404 :065,000,000,064,064,000,117
 52410 :192,048,003,000,192,000,109
 52416 :000,000,000,000,000,000,192
 52422 :000,000,000,000,000,000,198
 52428 :000,000,000,000,012,000,216
 52434 :000,051,000,000,004,000,009
 52440 :000,004,000,000,010,000,230
 52446 :000,010,000,000,010,128,114
 52452 :000,058,128,000,005,000,163
 52458 :000,049,000,000,001,000,028
 52464 :000,005,000,000,020,000,009
 52470 :000,021,000,000,004,128,143
 52476 :000,055,000,255,000,000,050
 52482 :000,000,000,000,000,000,002
 52488 :000,000,000,000,000,000,008
 52494 :000,000,048,000,000,204,010
 52500 :000,000,016,000,000,016,052
 52506 :000,000,160,000,002,160,092
 52512 :192,008,162,000,002,232,116
 52518 :000,000,080,000,000,084,202
 52524 :000,000,017,000,000,017,078
 52530 :000,000,065,000,001,001,117
 52536 :000,012,003,000,003,000,074
 52542 :192,000,000,000,000,000,254
 52548 :000,000,000,000,000,000,068
 52554 :000,000,000,000,000,000,074
 52560 :048,000,000,204,000,000,076
 52566 :016,000,000,016,000,000,118
 52572 :160,000,000,160,000,002,158
 52578 :160,000,002,172,000,000,176
 52584 :080,000,000,076,000,000,004
 52590 :064,000,000,080,000,000,254
 52596 :020,000,001,080,000,003,220
 52602 :016,000,000,220,000,255,101
 52608 :000,000,000,128,000,002,002

52614 :160,000,010,192,000,003,243
 52620 :048,000,012,252,000,063,003
 52626 :240,130,015,063,040,252,118
 52632 :015,020,240,000,130,000,045
 52638 :000,020,000,000,130,000,052
 52644 :002,000,128,002,000,128,168
 52650 :008,130,032,000,000,000,084
 52656 :000,000,000,000,000,000,176
 52662 :000,000,000,000,000,000,182
 52668 :000,000,000,000,000,000,188
 52674 :000,000,000,000,000,000,194
 52680 :000,000,000,000,000,000,200
 52686 :000,000,000,000,050,000,000
 52692 :140,060,130,060,207,020,061
 52698 :243,048,130,012,240,020,143
 52704 :015,128,130,002,130,000,117
 52710 :130,002,000,128,002,000,236
 52716 :128,008,130,032,000,000,022
 52722 :000,000,000,000,000,000,242
 52728 :000,000,000,000,000,000,248
 52734 :000,000,012,000,128,128,010
 52740 :000,002,160,000,002,000,168
 52746 :195,003,000,000,012,000,220
 52752 :000,015,240,130,015,015,175
 52758 :040,240,195,020,240,000,245
 52764 :130,003,048,020,000,000,229
 52770 :130,012,194,000,128,002,244
 52776 :048,128,008,128,032,000,128
 52782 :000,008,000,000,000,000,054
 52788 :000,000,000,000,000,000,052
 52794 :000,000,000,000,000,000,058
 52800 :000,000,000,000,000,000,064
 52806 :000,000,000,000,000,000,070
 52812 :192,000,192,000,048,003,255
 52818 :002,000,128,060,130,048,194
 52824 :192,020,243,048,002,012,093
 52830 :195,020,000,128,130,002,057
 52836 :142,000,142,002,032,000,162
 52842 :002,000,128,008,002,035,025
 52848 :000,000,000,000,032,000,144
 52854 :000,000,000,000,000,000,118
 52860 :000,000,000,000,189,255,056

Per collaborare a SUPERCOMMODORE

Se desiderate inviare alla rivista i vostri programmi o articoli, vi proponiamo una piccola 'guida', con lo scopo particolare di rendere più accurata la stesura dei manoscritti: seguendo questi consigli si accresceranno le probabilità che i vostri lavori vengano pubblicati.

1 l'angolo superiore sinistro della prima pagina dovrà contenere: nome, cognome, indirizzo, numero telefonico, codice fiscale, luogo e data di nascita.

2 l'angolo superiore destro della prima pagina dovrà contenere il tipo di computer al quale il lavoro si riferisce (C64 o C128), unitamente alle eventuali periferiche richieste.

3 il titolo dell'articolo, sottolineato, dovrà iniziare a circa due terzi in altezza della prima pagina.

4 l'articolo dovrà pervenire in Redazione sia registrato su supporto magnetico (formato SpeedScript o Easy Script per C64, oppure in formato ASCII MS-DOS o Macintosh) che stampato su carta.

5 analogamente i programmi, che accompagnano gli articoli dovranno essere inviati su supporto magnetico (floppy disk o cassetta); non ha importanza che venga fornito un listato su carta degli stessi.

6 nel caso i programmi siano in linguaggio macchina, dovranno essere specificati gli indirizzi iniziale e finale, e la procedura da seguire per il loro caricamento ed attivazione.

7 programmi in BASIC spezzati in più parti dovranno prevedere il caricamento in overlay da cassetta e (opzionalmente) da disco.

8 si tenga presente, che non potranno essere pubblicati listati troppo lunghi o riguardanti argomenti appena trattati sulla rivista.

9 non verranno presi in considerazione articoli o programmi non originali e già sottoposti ad altre case editrici.

10 il materiale pervenuto non sarà restituito, mentre verrà data comunicazione scritta o telefonica all'Autore soltanto nel caso in cui il lavoro sia stato ritenuto meritevole di pubblicazione.

11 il compenso spettante per la collaborazione prestata sarà commisurato alla complessità e all'interesse del lavoro (da un minimo di L. **50.000** a un massimo di L. **300.000**) e verrà corrisposto un mese dopo l'avvenuta pubblicazione sulla rivista.

Inviare articoli e programmi a:

SUPERCOMMODORE

Via Rosellini, 12
20124 Milano

e saremo lietissimi di pubblicare i contributi migliori.

La Redazione

Solarpix

di S. Edgeworth
trad. e adatt. di S. Colombo

Questa volta un programma veramente insolito: una simulazione astronomica del sistema solare, con tutti i Pianeti in moto, che permette anche di tracciare le orbite della Cometa di Halley e di altre, e soprattutto offre una visione affascinante del sistema di Pianeti in cui viviamo.

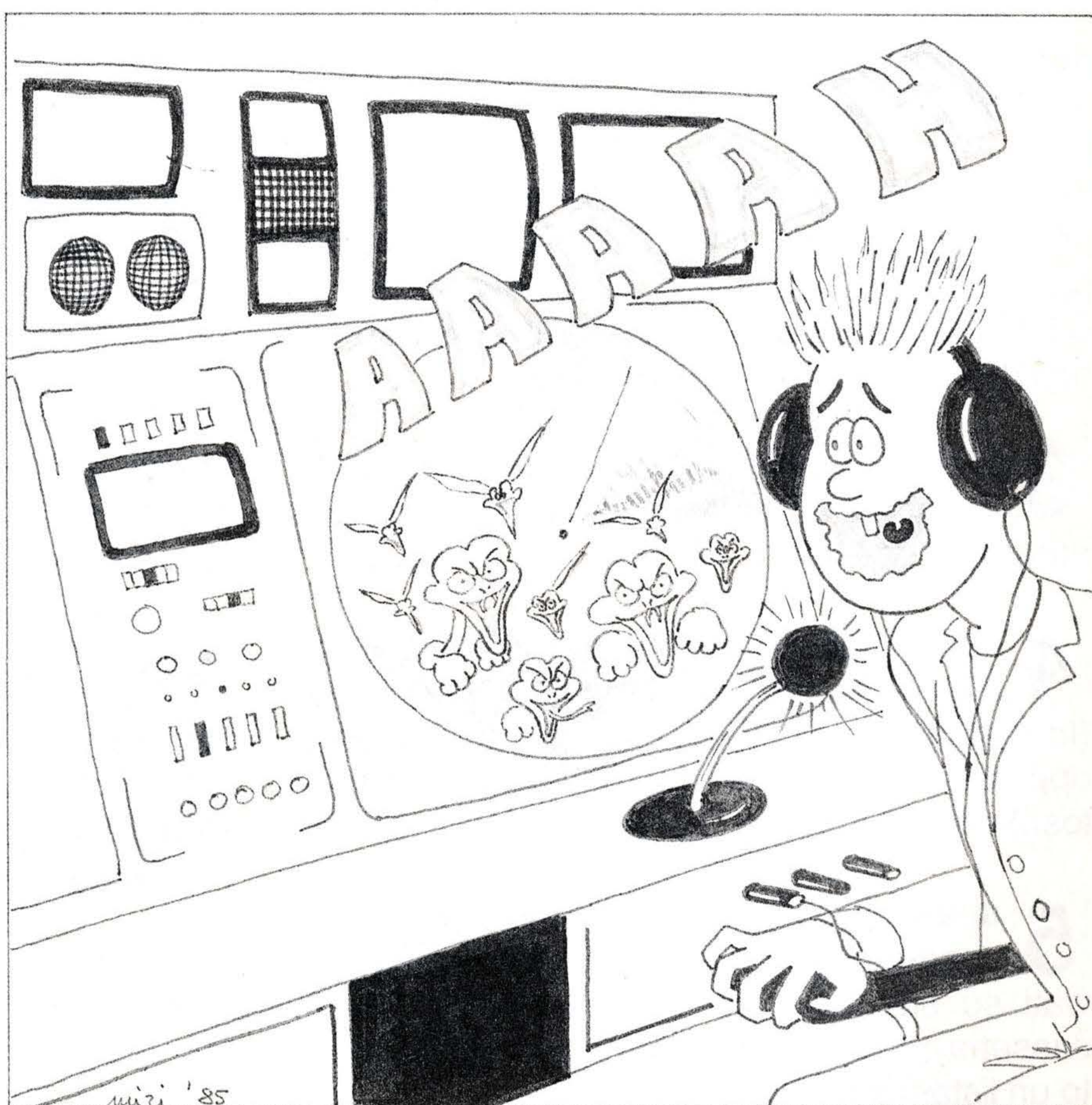
Il programma è diviso in quattro parti: Sole e Pianeti, in cui è simulato il movimento dei Pianeti attorno al Sole; Sole, Terra e Luna, che mostra come le fasi Lunari siano generate dall'orbita; Comete, in cui è simulata l'orbita della Cometa di Halley e di altre Comete; Altro, che mostra le principali caratteristiche di ciascun Pianeta.

Nella prima parte potete osservare le diverse orbite di cinque Pianeti alla volta: il Sole è fermo e si trova al centro dello schermo. Scegliete il nome del Pianeta più interno e il nome del Pianeta più esterno che desiderate osservare, e un anno da 1 a 2000. Premete G per porre i Pianeti in movimento, + per aumentarne la velocità, - per diminuirla, S per fermarne il movimento.

Il tempo trascorso viene visualizzato in anni nella parte destra dello schermo.

La Luna

La seconda opzione del menu mostra invece come l'orbita della Luna attorno alla Terra provochi le diverse fasi Lunari. La Terra è mostrata in orbita attorno al



Sole, mentre la Luna orbita attorno alla Terra: la distanza tra la Terra e la Luna è stata artificialmente aumentata di 30 volte per rendere tra loro distinguibili i due Pianeti.

Nell'angolo superiore destro dello schermo viene mostrata la fase Lunare corrente relativa alla posizione dei Pianeti, mentre il tempo trascorso viene visualizzato in giorni.

I controlli di cui disponete sono i tasti G, S, + e -, esattamente come nella descrizione precedente.

La Luna impiega 27.3 giorni per orbitare intorno alla Terra, mentre passano 29.6 giorni tra una Luna piena e la successiva. Se osservate attentamente lo schermo, potete comprendere il perché di questa differenza: per raggiungere la stessa posizione relativa al Sole la Luna deve completare qualcosa di più di una singola orbita attorno alla Terra.

Dal momento che l'orbita della Luna attorno alla Terra non è perfettamente circolare, normalmente non si verifica mai un perfetto allineamento tra Sole, Terra

e Luna. Nelle rare occasioni in cui questo allineamento si verifica, si ha il fenomeno dell'eclissi: se avviene quando la Luna è piena, sarà una eclissi di Luna, mentre se avviene a Luna nuova sarà una eclissi di Sole.

Le Comete

La terza opzione del menu mostra come appaiono le orbite delle Comete. I Pianeti e le Comete si muovono su orbite ellittiche, che appaiono come cerchi "schiacciati" su un diametro. Il parametro che misura di quanto il cerchio è "schiacciato" si chiama eccentricità, e oscilla tra zero (cerchio perfetto) e uno (cerchio completamente schiacciato, che si confonde con un segmento). Le eccentricità delle orbite dei maggiori Pianeti passano dal valore prossimo a zero di Venere a quello di 0.25 relativo all'orbita di Plutone.

Una Cometa la cui eccentricità orbitale fosse prossima a zero lascerebbe inesorabilmente il Sistema Solare per perdersi nello spazio, in quanto la forza di attrazione esercitata dal Sole non equilibrerebbe la forza centrifuga dovuta alla rotazione orbitale.

Inserite un valore tra 0 e 0.96 e premete RETURN. Ecco i nomi e le eccentricità

delle orbite delle principali Comete:

Cometa	Eccentricità
Faye	0.56
Finlay	0.70
Enke	0.85
Halley	0.96

Per osservare l'orbita della Cometa di Halley digitate 0.96R, dove la R indica che il moto della Cometa ha verso opposto a quello di rotazione dei Pianeti.

Si noti che la velocità di un Pianeta o di una Cometa varia in funzione della sua posizione lungo l'orbita: maggiore è la vicinanza con il Sole, maggiore è anche la velocità lungo l'orbita.

I dati relativi ai Pianeti

L'ultima parte del programma vi offre alcune nozioni interessanti circa il Sistema Solare. Dal menu potete scegliere un Pianeta, il Sole o la Luna e ottenerne tutti i dati principali; oppure potete scegliere una caratteristica - ad esempio la temperatura - e compararne i valori con quelli relativi a tutti e nove i Pianeti più il Sole e la Luna.

Dal momento che le distanze effettive tra i Pianeti e il Sole sono estremamente difficili da visualizzare, abbiamo aggiun-

to un valore veramente poco scientifico, ma di grande effetto: il "tempo guida" mostra infatti il tempo che si impiegherebbe a guidare da un dato Pianeta fino al Sole, tenendo una media costante di circa 95 Km/h. Per la Luna il "tempo guida" è relativo alla distanza Terra-Luna. Le masse dei Pianeti sono espresse in unità multiple della massa della Terra, dal momento che valori espressi in tonnellate sarebbero troppo ardui da comprendere (si pensi che la Terra pesa qualcosa come 6.588.000.000.000.000.000 tonnellate!). Le densità sono confrontate con quella dell'acqua (si noti che Saturno è meno denso dell'acqua), e le temperature sono le temperature medie della superficie dei Pianeti. Le varie lune sono elencate in ordine di distanza dai Pianeti attorno ai quali orbitano, iniziando dalle più vicine. Le lune contrassegnate da X non sono state ancora "battezzate" con un nome particolare.

Per i tecnici di astronomia ecco alcuni dettagli relativi a Solarpix: le orbite sono assunte come appartenenti tutte allo stesso piano. Per Plutone, invece, viene simulato il reale movimento ellittico. Lo schermo dei Pianeti è orientato con l'afelio di Plutone sulla destra, e le posizioni dei Pianeti sono calcolate dalla loro posizione reale del 27 Ottobre 1984.

Solarpix

```

10 POKE56,132:CLR:rem 216
20 POKE792,193:POKE648,132:rem 201
30 POKE56576,(PEEK(56576)AND252)OR1:rem 221
40 POKE53280,3:POKE53281,3:DIMS(255),C(255),M$(9,17):rem 42
50 BP=34808:V=53248:LC=49152:EY=1984.82:rem 156
60 PP=2*&:P$(0)="AC":P$(1)="DE":SA=1.1:rem 167
70 PRINT"{CLR}{ 8 GIU' }{BLU}"SPC(14)"ATTENDERE..." :rem 120
80 SP$=CHR$(32):X$=SP$+"Z=ESCI":rem 7
90 RE$=CHR$(18):RX$=CHR$(146):rem 78
100 T$(1)=RE$+"SOLE E{ 2 SPAZI}PIANETI"+RX$:rem 109
110 T$(2)=RE$+"SOLE,TERRA E LUNA"+RX$:rem 63
120 T$(3)=RE$+"COMETE"+RX$:rem 172
130 T$(4)=RE$+"ALTRO"+RX$:rem 115
140 FORJ=49152TO49240:READK:POKEJ,K:NEXT:rem 242
150 FORJ=34816TO35199:READK:POKEJ,K:NEXT:rem 252
160 FORJ=35520TO35775:POKEJ,0:NEXT:rem 58
170 FORL=1TO10:READJ,K:POKE35584+J,K:NEXT:rem 43

```

```

180 FORJ=0TO7:READK:POKEBP+J,K:NEXT:rem 18
190 FORJ=0TO7:READH(J):NEXT:rem 196
200 FORJ=0TO10:READF$(J),U$(J):rem 207
210 L=0:H=10:IFJ>4THENL=1:H=9:rem 210
220 FORK=LTOH:READZ(K,J):NEXT:NEXT:rem 238
230 FORJ=1TO9:D(J)=Z(J,5):rem 188
240 P(J)=Z(J,6):NEXT:rem 142
250 FORJ=0TO10:READN$(J):rem 156
260 IFJ<4ORJ=10THEN280:rem 51
270 FORK=1TOZ(J,10):READM$(J,K):NEXT:rem 176
280 NEXT:FORJ=1TO9:READR:R(J)=R/360:NEXT:rem 45
290 FORN=0TO255:M=N*2*&/256:rem 223
300 S(N)=SIN(M):C(N)=COS(M):NEXT:rem 159
310 FORJ=32TO36:K=74-J:rem 23
320 POKE251,J:POKE252,2:POKE253,K:rem 172
330 POKE254,2:SYS49175:NEXT:rem 122
340 FORJ=0TO7:POKEV+39+J,H(J):NEXT:rem 98
350 POKEV+23,0:POKEV+29,0:POKEV+16,128:rem 210
360 GOSUB550:POKEV+39,7:POKEV+40,3:rem 25
370 POKEV+32,3:POKEV+33,3:POKE646,6:rem 40

```



```

380 PRINT "{CLR}":SYSLC,4,14      :rem 184
390 PRINT "SOLARPIX":PRINT:PRINT :rem 109
400 FORJ=1TO4:PRINTTAB(8)J;T$(J):PRINT      :rem 144
410 NEXT:PRINTTAB(9)"Q ESCI"      :rem 181
420 GOSUB470:ONKGOTO630,1010,1280,1460      :rem 129
430 IFK$<>"Q"THEN420              :rem 105
440 POKE56576,PEEK(56576)OR3      :rem 87
450 POKE648,4:POKE792,71:GOSUB550      :rem 189
460 PRINT "{CLR}":POKE56,160:CLR:END      :rem 193
470 POKE198,0                      :rem 200
480 GETK$:IFK$=""THEN480          :rem 111
490 K=VAL(K$):RETURN              :rem 226
500 SYSLC,18,31:PRINT"G=VIA"      :rem 183
510 PRINTTAB(31)"S=STOP"          :rem 201
520 PRINTTAB(31)"+=VELOCE"        :rem 26
530 PRINTTAB(31)"-=LENTO"         :rem 225
540 PRINTTAB(30)X$:RETURN          :rem 71
550 POKEV+21,0:FORJ=0TO15         :rem 234
560 POKEV+J,0:NEXT:RETURN         :rem 132
570 SYSLC,23,30:PRINTX$          :rem 141
580 SYSLC,2,0:RETURN              :rem 255
590 GETK$:IFK$="S"THENFL=1        :rem 40
600 IFK$="G"THENFL=0              :rem 138
610 IFK$="-"THENM=M/2             :rem 170
620 RETURN                        :rem 120
630 POKEV+32,11:POKEV+33,0:POKE646,7 :rem 84
640 GOSUB550:PRINT "{CLR}"T$(1):GOSUB570 :rem 162
650 PRINT "[<7>]SELEZIONA I PIANETI      :rem 111
    { 3 SPAZI}(MAX5)"
660 PRINT:FORJ=1TO9:PRINTJ;N$(J):NEXT :rem 176
670 PRINT:PRINT"PIU' INTERNO?";      :rem 227
680 GOSUB470:IFK$="Z"THEN360        :rem 148
690 L=K:IFL<1THEN680               :rem 194
700 PRINTN$(L):PRINT"PIU' ESTERNO?";  :rem 237
710 GOSUB470:IFK$="Z"THEN360        :rem 142
720 H=K:IFH<LORH>9ORH-L>4THEN710:rem 253
730 PRINTN$(H):PRINT:X=1           :rem 249
740 FORJ=1TOH-L+1:X=X+2^J:NEXT      :rem 159
750 FORJ=LTOH:F(J)=D(J)*90/D(H) :rem 23
760 E(J)=F(J)*SA:NEXT:M=P(L)/50:ML=M*9 :rem 215
770 INPUT"ANNO INIZIALE (1-2000)";K$ :rem 14
780 SY=VAL(K$):IFSY<1ORSY>2000THEN630 :rem 34
790 PRINT "{CLR}{YEL}"T$(1):K=1     :rem 143
800 FORJ=LTOH:POKE646,H(K)          :rem 86
810 PRINTTAB(31)N$(J):K=K+1:NEXT:rem 161
820 POKE646,14:GOSUB500:SYSLC,7,31 :rem 4
830 PRINT"TERRA":PRINTTAB(31)"ANNI" :rem 167
840 E=.25:T=SY-EY:FL=1              :rem 47
850 POKEV,130:POKEV+1,148:POKEV+21,X :rem 151
860 FORJ=LTOH:A=T/P(J)+R(J)         :rem 129
870 A=INT((A-INT(A))*256):IFJ=9THEN890 :rem 181
880 X(J)=C(A):Y(J)=-S(A):GOTO910:rem 47
890 Z=1+E*C(A):X(J)=E+(E+C(A))/Z:rem 209
900 Y(J)=(E*E-1)*S(A)/Z            :rem 197
910 X(J)=INT(131+X(J)*E(J))         :rem 195
920 Y(J)=INT(149+Y(J)*F(J)):NEXT:rem 73
930 K=1:FORJ=LTOH:POKEV+2*K,X(J):rem 228
940 POKEV+2*K+1,Y(J):K=K+1:NEXT :rem 82
950 T$=STR$(INT((T+EY)*10)/10) :rem 134
960 SYSLC,9,30:PRINTT$SP$SP$      :rem 238
970 GOSUB590:IFK$="Z"THEN630        :rem 153
980 IFFLTHEN970                    :rem 145
990 IFK$="+"ANDM<MLTHENM=M*2       :rem 163
1000 T=T+M:GOTO860                 :rem 47
1010 POKEV+32,14:POKEV+33,0:POKE646,14 :rem 174
1020 PRINT "{CLR}"T$(2):PRINTTAB(31)"LUNA" :rem 233
1030 PRINTTAB(31)"FASE ":GOSUB500   :rem 143
1040 SYSLC,24,1:PRINT"(DIST.TERRA-LUNA MO :rem 227
    LTIPLICATA PER 30)";
1050 SYSLC,7,31:PRINT"TERRA"       :rem 208
1060 PRINTTAB(31)"GIORNI"          :rem 236
1070 E=84:D=E*SA:F=D*30/390:G=F/SA :rem 148
1080 M=0.3/365.25:T=0:H=13.3685 :rem 104
1090 POKEV+39,14:POKEV+40,12       :rem 84
1100 POKEV+12,141:POKEV+13,147 :rem 173
1110 POKEV+14,24:POKEV+15,80       :rem 78
1120 POKEV+21,195:FL=1             :rem 220
1130 A=T-INT(T):B=T*H:B=B-INT(B) :rem 121
1140 C=INT(12*(B-A)+.5-7)          :rem 2
1150 IFC<0THENC=C+12:GOTO1150      :rem 191
1160 A=A*PP:B=B*PP                 :rem 22
1170 X=INT(142.5+COS(A)*D)         :rem 164
1180 Y=INT(148.5-SIN(A)*E)         :rem 180
1190 W=INT(X+1+COS(B)*F)           :rem 98
1200 Z=INT(Y+1-SIN(B)*G)           :rem 102
1210 POKEV,X:POKEV+1,Y:POKEV+2,W:rem 12
1220 POKEV+3,Z:POKEBP+7,32+C       :rem 138
1230 SYSLC,9,30:PRINTINT(T*365.25) :rem 252
1240 GOSUB590:IFK$="Z"THEN360      :rem 192
1250 IFFLTHEN1240                  :rem 223
1260 IFK$="+"ANDM<.006THENM=M*2 :rem 245
1270 T=T+M:GOTO1130                :rem 95
1280 POKEV+32,5:POKEV+33,0:POKE646,5 :rem 87
1290 GOSUB550:PRINT "{CLR}"RE$T$(3):GOSUB5 :rem 145
    70
1300 INPUT"ECCENTRICITA'(0-.96)";K$ :rem 34
1310 E=VAL(K$):IFK$="Z"THEN360 :rem 197
1320 IFE<0ORE>.96ORLEN(K$)=0THEN1280 :rem 110
1330 FL=1:IFRIGHT$(K$,1)="R"THENFL=-1 :rem 126
1340 PRINT "{CLR}"RE$T$(3):GOSUB570 :rem 57
1350 PRINT"ECCENTRICITA'="E:SYSLC,2,27 :rem 108
1360 IFETHENPRINT"ACCELERAZIONE":rem 83

```


1370 M=0.01:T=0:F=65*SA:G=65*FL :rem	218	1800 DATA 253,174,32,158,173,32,170	
1380 POKEV,106:POKEV+1,148:POKEV+21,3		:rem	188
:rem	164	1810 DATA 177,96,162,6,6,251,38 :rem	2
1390 A=(T-INT(T))*PP:Z=1+E*COS(A)		1820 DATA 252,6,253,38,254,202,208	
:rem	138	:rem	138
1400 X=E+(E+COS(A))/Z:Y=(E*E-1)*SIN(A)/Z		1830 DATA 245,169,21,133,2,160,2:rem	30
:rem	237	1840 DATA 177,251,162,8,106,38,255	
1410 X=INT(107+X*F):Y=INT(149+Y*G)		:rem	149
:rem	193	1850 DATA 202,208,250,192,1,240,4	
1420 J=INT(A/&):POKEV+2,X:POKEV+3,Y		:rem	76
:rem	59	1860 DATA 152,73,2,168,165,255,145	
1430 IFE>0.1THENSYSLC,2,27:PRINTP\$(J)		:rem	150
:rem	239	1870 DATA 253,192,1,240,4,152,73:rem	37
1440 GETK\$:IFK\$="Z"THEN1280	:rem	1880 DATA 2,168,136,16,223,162,3:rem	40
1450 T=T+M:GOTO1390	:rem	1890 DATA 230,251,230,253,202,208,249	
1460 POKEV+32,3:POKEV+33,6:POKE646,3		:rem	29
:rem	89	1900 DATA 198,2,208,208,96	:rem
1470 PRINT"{CLR}"RE\$T\$(4)" MENU":SYSLC,4,		1910 DATA 0,15,0,0,0,192,0,0,48 :rem	211
0	:rem	1920 DATA 0,0,56,0,0,28,0,0,28	:rem
1480 GOSUB570:SYSLC,4,0	:rem	1930 DATA 0,0,14,0,0,14,0,0,7	:rem
1490 PRINT" S SOLE":PRINT" M LUNA"		1940 DATA 0,0,7,0,0,7,0,0,7	:rem
:rem	173	1950 DATA 0,0,7,0,0,14,0,0,14	:rem
1500 FORJ=1TO9:PRINTJ;N\$(J):NEXT:rem	19	1960 DATA 0,0,28,0,0,28,0,0,56	:rem
1510 SYSLC,4,0:FORJ=0TO10	:rem	1970 DATA 0,0,48,0,0,192,0,15,0,0	
1520 PRINTTAB(20)CHR\$(J+65)" "F\$(J):NEXT		:rem	53
:rem	211	1980 DATA 0,15,0,0,7,192,0,3,240:rem	14
1530 GOSUB470:IFK\$="Z"THEN360	:rem	1990 DATA 0,1,248,0,1,252,0,1,252	
1540 IFK\$="S"THENK=0:GOTO1590	:rem	:rem	64
1550 IFK\$="M"THENK=10:GOTO1590	:rem	2000 DATA 0,0,254,0,0,254,0,0,255	
1560 IFK>0ANDK<10THEN1590	:rem	:rem	46
1570 K=ASC(K\$)-65:IFK<0ORK>10THEN1530		2010 DATA 0,0,255,0,0,255,0,0,255	
:rem	135	:rem	49
1580 GOTO1720	:rem	2020 DATA 0,0,255,0,0,254,0,0,254	
1590 POKEV+32,3:POKEV+33,11:POKE646,3		:rem	48
:rem	137	2030 DATA 0,1,252,0,1,252,0,1,248	
1600 PRINT"{CLR}"RE\$N\$(K)RX\$;	:rem	:rem	50
1610 H=4:IFK=0ORK=10THENPRINT:GOTO1630		2040 DATA 0,3,240,0,7,192,0,15,0,0	
:rem	133	:rem	94
1620 H=10:PRINT" PIANETA#"K"DAL SOLE"		2050 DATA 0,15,0,0,15,192,0,15,240	
:rem	114	:rem	101
1630 PRINT:FORJ=0TOH	:rem	2060 DATA 0,15,248,0,15,252,0,15,252	
1640 PRINTF\$(J)TAB(18)Z(K,J)TAB(28)U\$(J)		:rem	212
:rem	0	2070 DATA 0,15,254,0,15,254,0,15,255	
1650 NEXT:IFK<10THEN1680	:rem	:rem	215
1660 PRINT"DIST.DALLA TERRA"TAB(19)"39090		2080 DATA 0,15,255,0,15,255,0,15,255	
0 KM."	:rem	:rem	218
1670 PRINTF\$(9)TAB(19)"6 MESI"	:rem	2090 DATA 0,15,255,0,15,254,0,15,254	
1680 PRINT:IFK<4ORK=10THEN1770	:rem	:rem	217
1690 FORJ=1TOZ(K,10):C=INT((J-1)/7)		2100 DATA 0,15,252,0,15,252,0,15,248	
:rem	234	:rem	207
1700 L=13+J-C*7:SYSLC,L,C*13	:rem	2110 DATA 0,15,240,0,15,192,0,15,0,0	
1710 PRINTM\$(K,J):NEXT:GOTO1770	:rem	:rem	190
1720 POKEV+32,3:POKEV+33,11:POKE646,3		2120 DATA 0,15,0,0,31,192,0,63,240	
:rem	132	:rem	100
1730 PRINT"{CLR}"RE\$F\$(K)RX\$TAB(19)U\$(K)		2130 DATA 0,127,248,0,127,252,0,127,252	
:rem	101	:rem	110
1740 L=0:H=10:IFK>4THENL=1:H=9	:rem	2140 DATA 0,255,254,0,255,254,0,255,255	
1750 SYSLC,5,0:FORJ=LTOH	:rem	:rem	119
1760 PRINTN\$(J)TAB(18)Z(J,K):NEXT		2150 DATA 0,255,255,0,255,255,0,255,255	
:rem	222	:rem	122
1770 GOSUB570:GOSUB470:GOTO1460	:rem	2160 DATA 0,255,255,0,255,254,0,255,254	
1780 DATA 32,13,192,132,251,32,13		:rem	121
:rem	78	2170 DATA 0,127,252,0,127,252,0,127,248	
1790 DATA 192,166,251,76,240,255,32		:rem	114
:rem	200		

2180 DATA 0,63,240,0,31,192,0,15,0,0	2550 DATA "DISTANZA DAL SOLE", "MILIONI KM
:rem 198	:rem 165
2190 DATA 0,31,0,0,255,192,3,255,240	2560 DATA 58,108,150,228,777
:rem 215	:rem 116
2200 DATA 3,255,248,7,255,252,7,255,252	2570 DATA 1427,2869,4497,5905
:rem 131	:rem 190
2210 DATA 15,255,254,15,255,254,31,255,25	2580 DATA "DURATA ANNO", "ANNI/TERRA"
5	:rem 93
:rem 21	2590 DATA 0.241,0.615,1,1.88,11.86
2220 DATA 31,255,255,31,255,255,31,255,25	:rem 135
5	:rem 14
:rem 20	2600 DATA 29.46,84,164,248
2230 DATA 31,255,255,15,255,254,15,255,25	2610 DATA "VELOCITA' ORBITALE", "KM/H"
4	:rem 118
:rem 23	2620 DATA 172000,125000,106000,86900
2240 DATA 7,255,252,7,255,252,3,255,248	:rem 232
:rem 135	2630 DATA 46600,35300,24100,19300,16000
2250 DATA 3,255,240,0,255,192,0,31,0,0	:rem 123
:rem 48	2640 DATA "INCLINAZ. ASSE", "GRADI"
2260 DATA 0,255,0,3,255,192,15,255,240	:rem 179
:rem 67	2650 DATA 0,179,23.5,25,3.1
2270 DATA 31,255,248,63,255,252,63,255,25	:rem 41
2	2660 DATA 26.7,97.9,28.8,0
:rem 31	:rem 16
2280 DATA 127,255,254,127,255,254,255,255	2670 DATA "TEMPO-GUIDA(90KM/H)", "ANNI"
:rem 188	:rem 53
2290 DATA 255,255,255,255,255,255,255,255	2680 DATA 75,139,193,295,1002
:rem 195	:rem 163
2300 DATA 255,255,255,127,255,254,127,255	2690 DATA 1840,3698,5797,7612
:rem 181	:rem 194
2310 DATA 63,255,252,63,255,252,31,255,24	2700 DATA LUNE,-
8	:rem 112
:rem 26	2710 DATA 0,0,1,2,16,17,5,3,1
2320 DATA 15,255,240,3,255,192,0,255,0,0	:rem 111
:rem 156	2720 DATA SOLE,MERCURIO,VENERE,TERRA,MART
2330 DATA 0,112,3,248,6,248,9,248,12,112	E,PHOBOS,DEIMOS
:rem 165	:rem 206
2340 DATA 64,192,67,192	2730 DATA GIOVE,METIS,ADRASTREA,AMALTHEA,
:rem 118	THEBE,IO,EUROPA,GANYMEDE,CALLISTO
2350 DATA 128,224,131,224,134,224	:rem 135
:rem 80	2740 DATA LEDA,HIMALIA,LYSITHEA,ELARA,ANA
2360 DATA 46,45,45,45,45,45,44,37	NKE,CARME,PASIPHAE,SINOPE
:rem 98	:rem 29
2370 DATA 7,3,4,14,8,6,7,12	2750 DATA SATURNO,ATLAS,X,X,JANUS,EPI METH
:rem 37	EUS,MIMAS,ENCELADUS,TETHYS,TELESTO
2380 DATA DIAMETRO,KM	:rem 75
:rem 0	2760 DATA CALYPSO,DIONE,X,RHEA,TITANO,HYP
2390 DATA 1390000,4980,12300,12750	ERION,IAPETUS,PHOEBE
:rem 147	:rem 42
2400 DATA 6750,141600,114200,51500	2770 DATA URANO,MIRANDA,ARIEL,UMBRIEL,TIT
:rem 133	ANIA,OBERON
2410 DATA 49900,2400,3500	:rem 147
:rem 205	2780 DATA NETTUNO,TRITONE,NEREIDE,X,PLUTO
2420 DATA MASSA,(TERRA=1)	NE,CARONTE,LUNA
:rem 192	:rem 252
2430 DATA 332830,0.055,0.815,1	2790 DATA 198,255,351,285,249,180,204,227
:rem 191	:rem 198
2440 DATA 0.107,318,95.2,14.5	
:rem 144	
2450 DATA 17.2,0.01,0.012	
:rem 181	
2460 DATA "DENSITA'", "(ACQUA=1)"	
:rem 243	
2470 DATA 1.4,5.4,5.2,5.5,3.9,1.3	
:rem 70	
2480 DATA 0.7,1.2,1.7,1.4,3.4	
:rem 124	
2490 DATA "PER.ROTAZIONE", "GIORNI/TERRA"	
:rem 226	
2500 DATA 25,58.9,243,1,1.03,0.404	
:rem 122	
2510 DATA 0.444,0.67,0.77,6.39,27.3	
:rem 189	
2520 DATA TEMPERATURA,CENTIGRADI	
:rem 51	
2530 DATA 5500,320,480,22,-23,-150	
:rem 116	
2540 DATA -180,-206,-220,-240,-23	
:rem 54	

Biblioteca Jackson per il lettore di Commodore 64 e 128

Lire

Commodore 64	8.500
Easy Script	8.500
Voi e il vostro Commodore 64	22.000
Commodore 64. Il BASIC	26.000
Il libro dei giochi del Commodore 64	13.000
Il libro dei giochi del Commodore 64 (con cassetta)	24.000
Commodore 64. I file	17.000
Facile guida al Commodore 64	13.500
Il grande libro dei giochi (con cassetta)	24.000
Commodore 64. La grafica e il suono (con cassetta)	34.000
Programmazione del 6502	30.000
Applicazioni del 6502	15.500
Giochi con il 6502	19.500
Statistica a una dimensione con il Commodore 64	17.000
Matematica e Commodore 64 (con cassetta)	24.000
Adventure e Commodore 64 (con cassetta)	35.000
Il linguaggio macchina del Commodore 64 (con cassetta)	29.000
Grafica e Commodore 64 (con cassetta)	15.000
Commodore 64 a scuola	18.000
Tecniche di programmazione sul Commodore 64	16.500
Il sistema Totomac (con cassetta)	24.000
Impara il bridge con il computer C64 (con dischetto)	50.000
Romanzo rosa con il C64 (con dischetto)	40.000
Commodore Avventure (con cassetta)	20.000
La fisica con il computer: la dinamica (con cassetta)	35.000
Commodore 128 oltre il manuale	29.000
Programmi per Commodore 128 (con cassetta)	27.000

FAI IL PIENO DI INFORMATICA

I più venduti

- 1) Commodore 64
- 2) Easy Script
- 3) Commodore 64. Il BASIC
- 4) Programmi per 128
- 5) Commodore 64. I File

161 Boris Allan Grafica e Commodore 64

Pagine 152 L. 15.000
Cod. 573D

L'argomento maggiormente trattato in questo libro è lo sviluppo di un sistema di "turtle graphics" ad alta risoluzione per il Commodore 64. Il testo è uno strumento che permette di realizzare risultati grafici di buona qualità sfruttando le possibilità nascoste di questa macchina. Infatti, sia il manuale d'uso che la guida di riferimento risultano

carenti ed oscure a chi si avvicini all'uso dell'home computer senza precedenti conoscenze informatiche. Aiutare il lettore a superare queste situazioni è l'obiettivo dell'Autore, che si sforza di mantenere sempre l'esposizione delle tecniche usate su di un piano il più possibile comprensibile per il neofita. I concetti iniziali sono espressi secondo il diffuso linguaggio dei Turtle Graphics, nucleo del linguaggio LOGO. L'insieme delle procedure presentate, dedicate a singole operazioni grafiche, costituisce un organico di strumenti per ideare immagini sempre più complesse, utilizzabili anche all'interno di programmi ideati dal lettore.

È necessaria una conoscenza della programmazione Basic e uno spiccato interesse verso le applicazioni grafiche anche su periferiche dedicate.

104 Rita Bonelli Commodore 64. Il Basic

Pagine 324 L. 26.000
Cod. 348D

L'accurata esposizione del linguaggio BASIC e l'ampiezza, degli esempi sono una costante dei libri di Rita Bonelli, autrice di fama consolidata e conosciutissima tra gli appassionati di informatica. Quest'opera diventa così indispensabile agli acquirenti del C64 che desiderino realmente imparare ad utilizzarlo, scrivendo in proprio i programmi BASIC. È così colmata la lacunosa manualistica di cui il C64 è dotato, e finalmente i possessori di Commodore non dovranno più rivolgersi a case editrici straniere o a malfatte traduzioni per acquisire tutte le conoscenze utili all'uso intelligente del loro computer.

Proprio l'alta qualità ha reso possibile il successo e la diffusione di questo testo ormai "celebre" tra tutti i possessori e gli utenti del Commodore 64.



**GRUPPO EDITORIALE
JACKSON**

DIVISIONE LIBRI

Tua aria di novità

FINAL CARTRIDGE

Il primo sistema operativo esterno per Commodore 64



Nel 1985 ha vinto il premio Oskar per la migliore utility dell'anno. Nel 1986 supera se stessa e si impone come la cartuccia più completa in circolazione.

Vero e proprio **sistema operativo esterno (32K di ROM)** non occupa neppure un byte di memoria e fa compiere un bel salto di qualità al tuo 64.

Turbodisco - 6 volte più veloce in lettura, 8 volte più veloce in scrittura.

Turbonastro - 10 volte più veloce, anche con i files. Usa i normali comandi Commodore ed è compatibile con il turbo standard.

Freeze Frame - Permette di congelare in memoria praticamente tutti i programmi per un back-up totale e automatico su disco/cassetta e per la riproduzione su carta delle schermate in Hi-Res. Pagina piena con 12 gradazioni di grigio. In modalità training disabilita il riconoscimento della collisione fra sprites e vi garantisce... l'immortalità.

Interfaccia avanzata Centronics - Permette di stampare tutti i simboli grafici e i codici di controllo Commodore su qualsiasi stampante standard centronics. Molto importante per i listati.

24K di RAM in più per i programmi Basic - Due nuovi comandi, "Memory read" e "Memory wri-

te", possono spostare 192 bytes con la velocità del Linguaggio Macchina ovunque nell'ambito dei 64K di RAM del Commodore 64. Utilizzabili con stringhe e variabili.

Comandi Basic 4.0 - Come Dload, Dsave, Dappend, Catalog, cc.

Basic Toolkit - Con Auto, Return (compresi Goto e Gosub), Find, Help, Old, ecc.

Tasti funzione preprogrammati - Run, Load, Save, Catalog, comandi per gestione disco, List (rimuove tutte le eventuali protezioni).

Nuove funzioni da tastiera - Per cancellare una parte di una riga, fermare e riprendere un listato, spostare il cursore nell'angolo in basso a sinistra. Con il comando Type potete usare la stampante come una macchina da scrivere.

Monitor L/M esteso - Con assembler e disassembler. Non risiede in memoria.

Pulsante di reset - Sempre molto utile.

Interruttore On/Off - Per evitarvi il fastidio di togliere e rimettere la cartuccia.

VOICEMASTER

Processore vocale e musicale per:

Commodore 64 e 128 (in modo 64), Atari 800/800XL/65XE/130XE Apple II+/IIe/IIc (il II+ richiede 64K e paddle port adapter)

Nuova versione di software con Editor per modificare le caratteristiche dei suoni da riprodurre per una fedeltà ancora migliore.

Nuovo prezzo ribassato a 199.000 lire. E per II+/IIe puoi avere in opzione a 89.000 lire la scheda **Soundmaster**, che consente all'Apple le stesse prestazioni musicali di Commodore e Atari.

Programmi parlanti?

I tuoi programmi possono parlare con la tua voce, in qualsiasi lingua e con qualsiasi accento. Usa il comando **LEARN** e parla nel microfono; con il comando **SPEAK** il computer restituirà all'altoparlante quello che hai detto.

Il manuale riporta numerosi esempi di come aggiungere la voce ai programmi.

Gli ordini si danno a voce!

Ti piacerebbe dare ordini al computer con la tua voce? Semplicissimo! Con il comando **TRAIN** gli comunichi le parole o le frasi da riconoscere e con il

comando **RECOGNISE** ottieni che le capisca e si comporti di conseguenza.

Un po' di musica per rilassarti

Con il software del Voicemaster puoi comporre e suonare semplicemente canticchiando o fischiettando. Non importa se non sai leggere la musica.

La tua voce o il tono del fischio scriveranno le note per te!

Mentre canti o fischi le note appaiono sullo schermo: le puoi correggere, riascoltare e, quando sei soddisfatto, puoi stampare lo spartito. In modo performance puoi cambiare le ottave, le chiavi o aggiungere accordi.

Questo prodotto, a questo prezzo, può confrontarsi unicamente con apparecchiature di costo pari a qualche milione. Vuoi una prova? Richiedici a sole 29.000 lire il **dischetto parlante dimostrativo** che non necessita di hardware aggiuntivo e contiene alcuni vocabolari generici utilizzabili nei tuoi programmi. Insieme al dischetto riceverai un manuale esplicativo in Italiano, utile anche come introduzione al Voicemaster.

Il prezzo pagato per il dischetto dimostrativo ti sarà interamente rimborsato nel caso di successivo acquisto del Voicemaster.

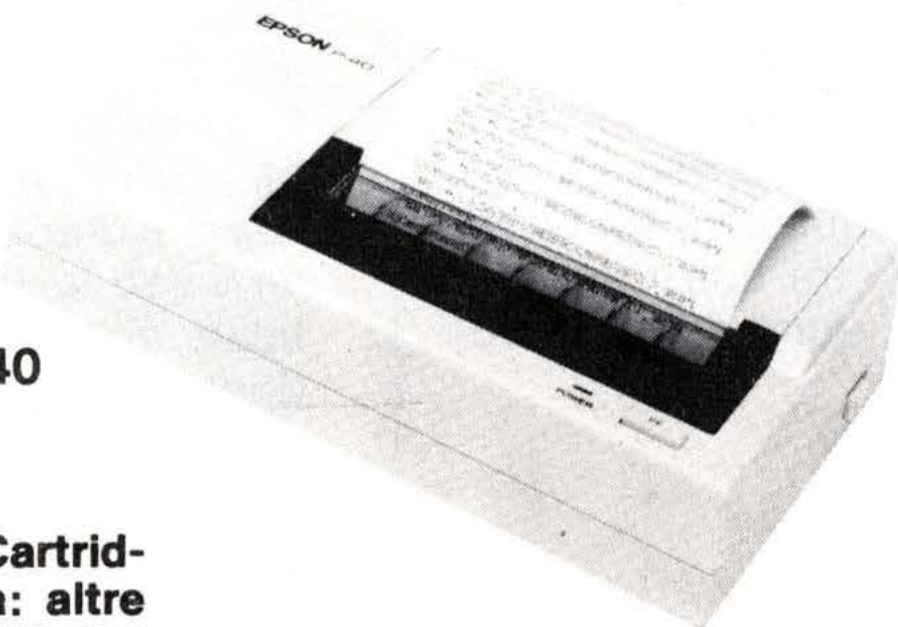
EPSON P40

ECCEZIONALE!
Stampante EPSON P40
all'incredibile prezzo
di lire 230.000.

E per gli amici del Final Cartridge una ventata di follia: altre 71.000 lire di sconto (159.000 anziché 230.000) solo se acquistata insieme alla cartuccia.

- EPSON P40 portatile con alimentazione a batteria e a rete
- Stampa a matrice di punti

- Grafica
- 80 colonne in compresso (larghezza della carta 112 mm)
- Collegabile al C64 tramite il cavo Centronics fornito in opzione con The Final Cartridge

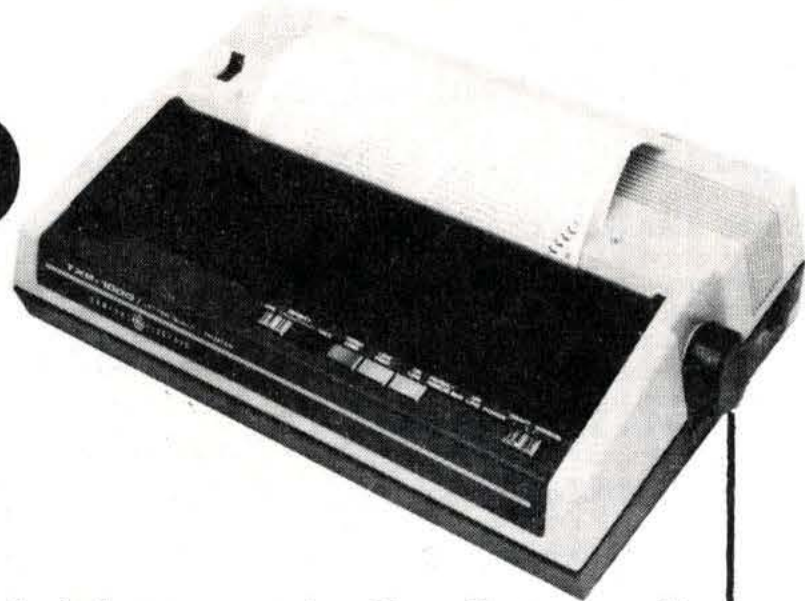


in casa MICROSTAR

TXP-1000

PREZZO IRRIPETIBILE
Standard centronics per
MSX/Apple/Amstrad/IBM
PC 399.000 (listino 500.000)
Interfaccia diretta Commo-
dore/Atari 449.000 (listino
570.000)

Questa **stampante universale, letter quality**, può essere colle-
gata a qualsiasi computer dota-
to di interfaccia standard centro-
nics tramite apposito cavo (non
compreso).
Per gli utenti Commodore/Atari
la TXP-1000 viene già fornita con
interfaccia diretta e cavo di col-
legamento. **Non è richiesto**



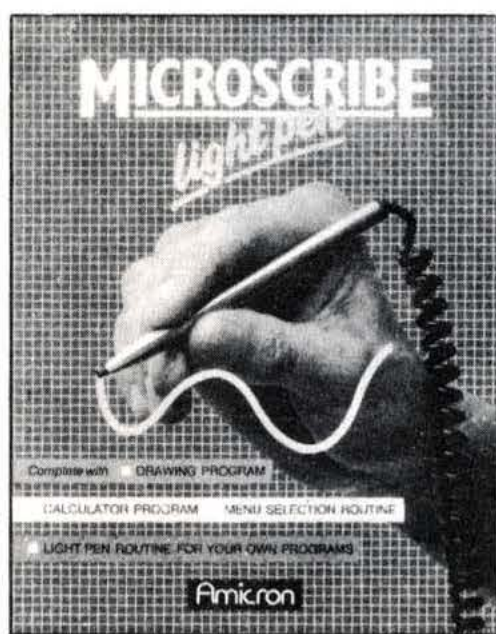
hardware aggiuntivo. Il prezzo di
lancio sfida le leggi mercato. Fin-
chè dura.

- Testina di stampa a 16 aghi. Car-
rello standard 80 colonne.
- Silenziosissima.
- Foglio singolo e modulo conti-
nuo. Carta normale o termica.
- Due velocità: 50 o 25 caratteri/
secondo.
- Qualità di stampa controllabile
tramite il cursore "Density".

LIGHT PEN MICROSCRIBE

Costruita in acciaio inossidabile,
Microscribe è in grado di pilotare
in maniera molto precisa gli spo-
stamenti del cursore **anche re-
stando a 5 cm di distanza dallo
schermo.**

Il software per disegnare in alta
risoluzione si basa sul principio
del menù ad icone e permette,
tra l'altro, il tracciamento a mano
libera, lo zoom, la cancellazione,
il salvataggio delle immagini su
disco o cassetta, l'uso della ta-
stiera per introdurre testo. Tutto
normale, a parte la qualità.
Assolutamente **eccezionali so-
no invece altri due programmi in**



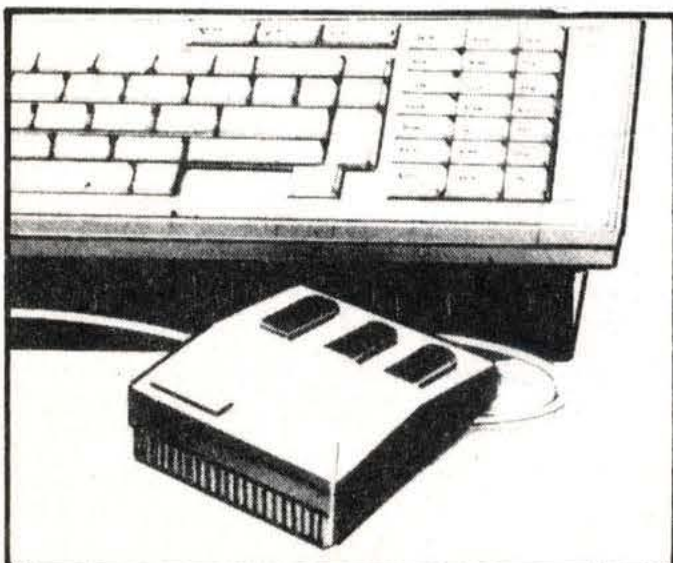
dotazione: una calcolatrice su
schermo per fare direttamente
tutti i vostri conti e un sistema
per utilizzare la light pen dentro i
programmi BASIC.

**Il prezzo? Solo 95.000 per la
versione su cassetta e 98.000
per il floppy.**

MEGAMOUSE

**L'unico mouse degno
del Commodore**

Il software è completamente
guidato tramite **icone**, ossia sim-
boli grafici racchiusi in un qua-



drato a rappresentare le diverse
funzioni.

Per selezionare una funzione ba-
sta far scivolare il mouse sul pia-
no della scrivania fino a portare il
cursore sopra la icona prescelta
e quindi premere brevemente il
pulsante adatto.

L'uso di Megamouse risulta tal-
mente immediato e naturale da
non poterne fare a meno: com-
puter art, CAD, disegno indu-
striale, grafici di ogni tipo sono
solo alcune delle applicazioni
possibili. E i risultati faranno im-
pallidire i fratellini maggiori a 32
bit.



OAK

Fa bene sedersi bene

tevole. La maggior parte del cari-
co è sopportata dai femori e la
**colonna vertebrale rimane in
posizione corretta.** Oltre a ciò,
OAK è molto più bella di una nor-
male sedia e il suo prezzo è sba-
lorditivo: solo 119.000 lire. **E se
te ne servono due risparmi
24.000 lire: 214.000 lire invece
di 238.000.**

Creata da un team di designers
in collaborazione con medici e fi-
sioterapisti e realizzata in legno
di rovere, **OAK distribuisce il pe-
so del corpo in maniera bilan-
ciata rispetto al baricentro**, ri-
sultando estremamente confor-



Non perdere tempo.
Ritaglia e spedisce
oggi stesso a:
Jackson S.p.A.
Via Rosellini, 12
20124 MILANO

INVIATEMI (barrare le caselle opportune)

- | | |
|--|--|
| Voicemaster (specificare computer _____) | ☐ L. 199.000 |
| Dischetto parlante (specificare computer _____) | ☐ L. 29.000 |
| Soundmaster per Apple II+/Ile | ☐ L. 89.000 |
| The Final Cartridge | ☐ L. 147.000 |
| Cavo parallelo centronics (opzionale) | ☐ L. 36.000 |
| Stampante EPSON P40 (centronics) | ☐ L. 230.000 |
| EPSON P40 + Cavo parallelo + Final Cartridge | ☐ L. 342.000 |
| Stampante G.E. TXP-1000 | |
| ☐ L. 399.000 (centronics) | ☐ L. 449.000 (Commodore/Atari) |
| Lighpen Microscribe | ☐ L. 95.000 (cassetta) ☐ L. 98.000 (floppy) |
| Sedia OAK | ☐ L. 119.000 (singola) ☐ L. 214.000 (coppia) |
| Megamouse | ☐ L. 125.000 |

Tutti i prezzi comprendono IVA e spese di spedizione in contrassegno postale

Nome /Cognome

Indirizzo

CAP/Città

Cod. Fisc. (solo se si desidera fattura)

Data

Firma

La spedizione sarà effettuata direttamente dall'importatore MICROSTAR s.r.l.
Solo per informazioni tecniche: (02) 6597693

Tutto il materiale è coperto da 12 mesi di garanzia.

input.output

**Piccoli
annunci**



Vendo videogame per C 64. Per avere la lista inviare L. 550 in francobolli. Superofferta: una cassetta con giochi del calibro di Donkey Kong, Calcio II, Baseball, One to one, Popeye, Speed King e Ghostbuster, tutti con Turbo Tape. Marcello Crippa - Viale Giulio Cesare, 28 - 22100 Como - Tel. 031/261014

Vendo software su disco e cassetta per C 64, utility e giochi. Ultime novita', Frankie goes to Hollywood, Space station, Rambo II, Pascal, Forth, Logo, ecc. Raffaele Miele - Via del Gran Sasso, 15 - 80144 Napoli - Tel. 081/7113229

Cedo software su dischi in cambio di francobolli, monete e cartamoneta per collezione. Rispondo a tutti. I dischi a vostre spese. Salvatore Careddu - Via G. Deledda, 111 - 07100 Sassari - Tel. 079/260336

Compro C 64, possibilmente con registratore. Offro L. 160.000 al massimo. Vittorio Schrru - Via Cupa Tozzole, 3 - 80100 Napoli - Tel. 081/7414646

Cambio programmi C 64 con i manuali di: Blazing paddles, M. Interrupt, McTutor, Doodle II, Logo, Zoom-Pascal, G-Pascal, Pilot, Music studio, Master 64. Stefano Nose' - Via Magenta, 17 - 37069 Villafranca - Tel. 7903129

C 16 Computer Club: disponiamo di oltre 150 titoli

per C 16, che vendiamo a L. 4.000 cadauno oppure 2.000 titoli per C 64, sempre allo stesso prezzo. Per informazioni rivolgersi a: Dorian Battagin - Via Firenze, 20 - 13060 Valdengo (VC)

Vendo 4 cartucce per C 64: Jack Attack, Visible Solar System, Jupiter Lander e Kickman originali della Commodore a L. 10.000 l'una. Annuncio sempre valido. Andrea Tura - Via Camelia, 17 - 20095 Cusano Milanino (MI) - Tel. 02/6133020

Per C16 vendo e cambio programmi di giochi, utility e didattici in genere. Tratto preferibilmente in zona Milano, telefonare ore pasti. Giovanni Marchesoli - Via Ungheria, 15 - 20138 Milano - Tel. 02/503173

Vendo moltissimi programmi listati, utility, videogame; ultimissime novita': Impossible Mission, Star League, Baseball, Summer Games. Richiedere lista. Generoso Fummo - Via Vincenzo Schiavo, 6 - 84100 Salerno - Tel. 089/355369

Vendo consolle Intellivision piu' 5 cassette gioco (Soccer, Tennis, Snafu, Shark-Shark, Popeye) a L. 219.000. Solo zona Triveneto. Telefonatemi ore pasti. Luca Ardiri - Via Verona, 12 - 37022 Fumane (VR) - Tel. 045/7701732

Vendo e cambio programmi per C 64. Ultime novita' e oltre 3.000 programmi solo su disco; i prezzi vanno da L. 2.000 a L. 4.000. Cambio solo a livello di novita' 1985-'86. Claudio Bacci - Via dei Pelaghi, 165 - 57100 Livorno - Tel. 0586/852551

Vendo per il tuo C 64 2.000 programmi a prezzi imbattibili; tutte le ultime novita'. Massima serietà annuncio sempre valido. Telefonare dalle ore 19.00 alle ore 21.00. Maurizio Anfosso - Via Simonetti, 4 - 14018 Roatto d'Asti (AT) - Tel. 0141/938118

Per C 64 vendo giochi magnifici (Calcio, Pac Man, Tarzan, QBert, Baseball, ecc.) a partire da L. 500. Per richiedere la lista e i prezzi scrivete a: Ivan Maccagnan - Via Jacopo Facen, 9 - 32033 Lamon (BL)

Vendo C 16 piu' registratore, joystick e 6 giochi a L. 220.000. Giuseppe Sambiase - Via Casilina Km 23, 400 - 00040 Roma provincia - Tel. 06/9462226

Per VIC 20 vendo giochi e programmi. Dispongo di oltre 150 titoli, per lista gratuita scrivere o telefonare ore pasti. Remo Bibbiani - Via Pessagno, 17 - 57018 Vada (LI) - Tel. 0586/787256

Cerco i seguenti giochi per C 64: F.B. Boxing (Elite), Summer Games II (Epyx), Tour de France (Activision), On court tennis (Activision), Boxing (Activision), Scooby Doo (Elite), Rambo (Ocean), Winter Games (Epyx), Street Hawk (Ocean). Solo su cassetta. Antonio Procopio - Via Nazionale, 160 - 88060 Rosolato Marina (CZ) - Tel. 0967/81021

Vendo giochi per C 16 e plus/4 tra cui Calcio, Skramble, Pac Man e Tennis, molti dei quali turbizzati a L. 1.500 cadauno. Telefonare o scrivere a: Rino De Vita - Via E. Gesu', 9 - 88100 Catanzaro - Tel. 0961/26292

Vendo C 64 (5 mesi di vita) causa acquisto moto. Vendo inoltre registratore, piu' di 100 programmi (utility e giochi), la guida all'uso e numerose altre riviste a L. 450.000 trattabili. Luca Busato Ruffini - Via Zelli di Cognola - 38100 Cognola - Trento - Tel. 0461/982612

Vendo VIC 20 piu' espansione 16 Kbyte, 30 cassette, quattro cartridge, libri manuali. Tutto in ottimo stato a L. 250.000. Telefonare nel pomeriggio. Roberto Antonioni - Via Canton - 00144 Roma - Tel. 06/6888222

input-output

**Piccoli
annunci**



Vendo cassetta giochi per C 64, con 10 giochi a scelta, a L. 15.000 (Buck Rogers, Pole Position, Soccer, ecc.). Vendo C 16 con registratore, joystick e oltre 100 programmi a L. 450.000 trattabili. Stefano Santambrogio - Via Oberdan, 12 - 20038 Seregno (MI) - Tel. 0362/235675

Per C 16 vendo "Introduzione al Basic" parte I a L. 30.000 (due cassette con manuale in italiano). Vendo inoltre cartridge per VIC 20 a L. 10.000. Vendo software su disco per C 64 (possiedo piu' di 600 programmi). Eventualmente sono disposto a cambiarne. Idelfo Borgo - Via Ponti Alti, 58 - 36030 Zugliano (VI) - Tel. 0445/872785

Vendo per C 64 cassetta con 35 programmi (es. Summer Games, Robb, Raid over Moscow, Pit Stop II, Bruce Lee, S.S. Challenge, Stealth, Designer's pencil, Quill adventure writer. Stefano Longhi - Via Verdi, 25 - 20053 Muggio' (MI)

Per C 64 cambio software preferibilmente su disco. Aspetto la lista. Rispondo a tutti solo se per cambio. Annuncio sempre valido. Giovanni Pugliese - Via A. Volta, 93 - 74100 Taranto - Tel. 099/413769

Vendo e cambio programmi per C 64 su cassetta. Ultima novita': Rambo II. Inoltre: Kung fu, Masters, Soccer 4, Impossible Mission II e altri eccellenti giochi. Rosario Mastantuono - Via Raffaele Cettieri - 84100 Salerno - Tel. 089/333168

Compro qualsiasi programma per C 64. Inviare la lista. Cristiano Cravanzola - Via Prof. Oliva, 16 - 12011 Borgo S. Dalmazzo (CN)

Vendo tavoletta grafica "Graphic Commander" originale Commodore a L. 180.000, stampante MPS 802 a L. 430.000. In regalo 30 programmi, 8 utility e una sorpresa. Simone Badioli - Via Torconca, 1400 - 47048 S. Giovanni in Marignano (FO) - Tel. 0541/760990

Cambio numerosi programmi per C 64: giochi, utility, gestionali e grafici in alta risoluzione. Inviare la vostra lista, rispondo a tutti. Walter Di Pace - Via Dante, 32 - 47041 Bellaria - Tel. 051/44090

Vendo, cambio, compro giochi su cassetta o disco. Possiedo i giochi piu' famosi e interessanti come: The hobbit II, Rambo II, S15 Strike Eagle, Conan, Boxe, ecc. Telefonare in ore pomeridiane. Alessandro Di Dio Rosso - Via Chiusi - 00139 Roma - Tel. 06/810788

Vendo il miglior software per C 64 a prezzi stracciati: Catch, Dragons Lair, Asterix, The Last V8, Back to the future, Fast hackem, Di/Sector arsenal, ecc. Tutti programmi su disco. Marco Zamponi - Via XX Settembre, 48 - 63014 Montegranaro (AP) - Tel. 0734/889914

SUPERCOMMODORE

CEDOLA DI ORDINAZIONE RIVISTE ARRETRATE CON CASSETTA ALLEGATA da compilare e spedire in busta chiusa a Gruppo Editoriale Jackson - Via Rosellini, 12 - 20124 Milano - Tel. 6880951/2/3/4/5

Vogliate inviarmi i numeri

di SUPERCOMMODORE con cassetta Anno

Al prezzo di L. 15.000 cad.

☐ Contanti allegati ☐ Assegno allegato n°

☐ Ho spedito l'importo a mezzo vaglia postale

☐ Ho versato l'importo sul ccp. n° 11666203 intestato a Gruppo Editoriale Jackson - Milano

☐ Pagherò in contrassegno al postino al ricevimento

BUONO D'ORDINE PER RIVISTE ARRETRATE

Anche se l'ordine riguardasse la sola cassetta, questa verrà comunque inviata insieme alla rivista al prezzo indicato.

Nome

Cognome

Via

Città C.A.P. Prov.

Se richiesta fattura:

Cod. F. e P. Iva

Data

Firma

Per i residenti all'estero - pagamento anticipato (vaglia o versamento su ns. ccp)

Ogni settimana
in edicola

VIDEO BASIC



Venti fantastiche lezioni settimanali su cassetta, per apprendere in modo facile, interattivo e divertente, il BASIC, i principi di funzionamento dell'hardware e le tecniche di programmazione degli home computer più diffusi. Ogni lezione contiene esercizi di familiarizzazione con la macchina e con il suo linguaggio, in più alcuni giochi di crescente difficoltà, analizzati nella loro struttura.

OGGI IN 5 VERSIONI

C-64
& **C-128**

MSX

C-16
& **PLUS/4**

VIC-20
Spectrum

**PER DIALOGARE
DAVVERO E SUBITO
COL TUO COMPUTER**



**GRUPPO EDITORIALE
JACKSON**
DIVISIONE GRANDI OPERE

Ogni settimana in edicola

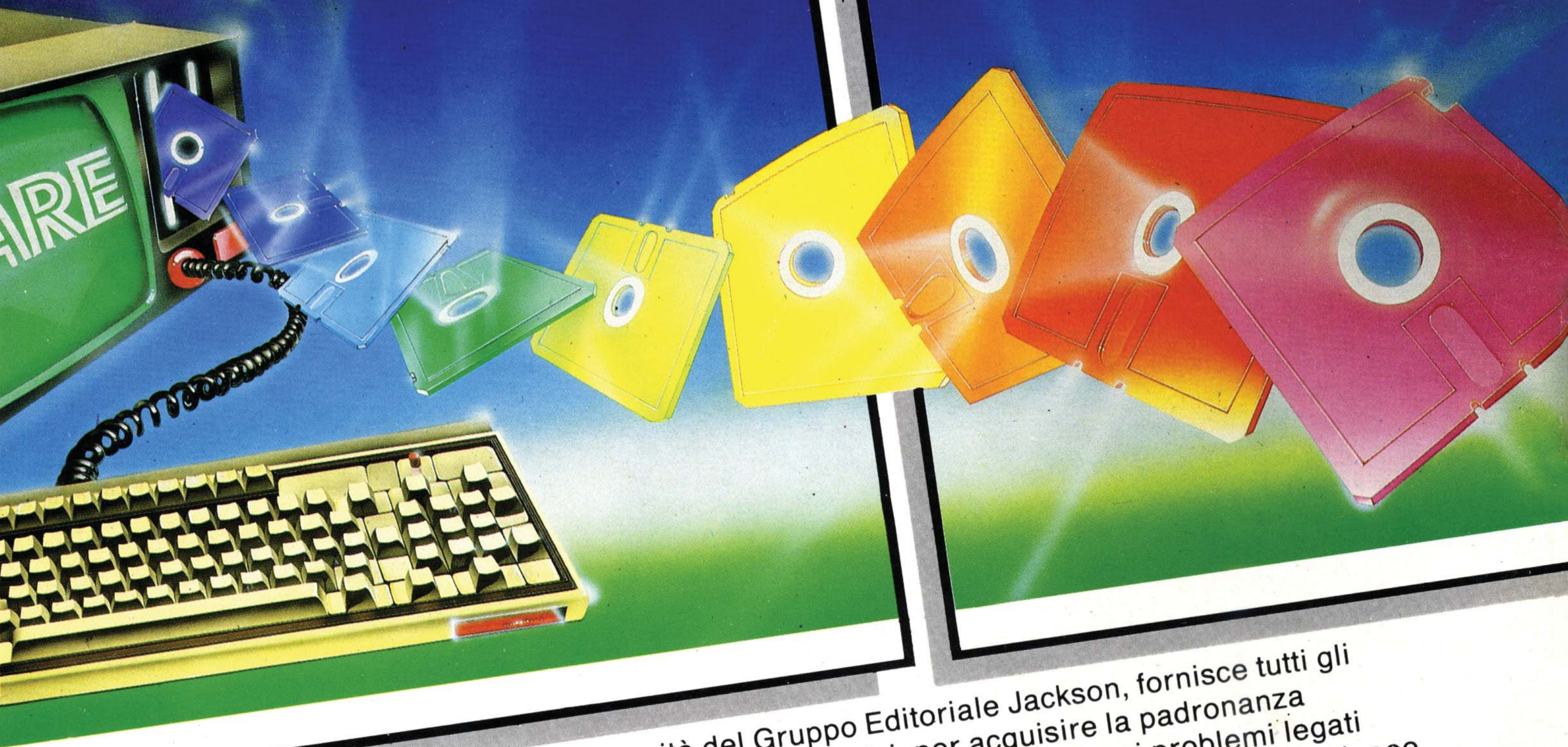
SOFTWARE

**APPLICAZIONI
LINGUAGGI
SISTEMI OPERATIVI
E PROGRAMMAZIONE
DEI PERSONAL COMPUTER**

**WORD PROCESSOR • PASCAL • FORTRAN •
DATA BASE • COBOL • "C" ...
FOGLI ELETTRONICI • MS DOS • C/PM •
COMPUTERGRAFICA • XENIX • UNIX •
BASIC • LOGO • UCSD**

Software si compone di 52 fascicoli settimanali,
da rilegare in 5 splendidi volumi:
**BASIC I E II • SISTEMI OPERATIVI •
LINGUAGGI • APPLICAZIONI •**

**1° E 2°
FASCICOLO
A SOLE
Lire 2.200**



Software, ultimissima novità del Gruppo Editoriale Jackson, fornisce tutti gli strumenti teorici, ma soprattutto pratici, per acquisire la padronanza completa del personal computer. Per risolvere, finalmente, i problemi legati all'uso pratico della macchina; per comprenderne le soluzioni applicative più idonee. Ottimo per il principiante, che intende accedere al mondo dell'informatica dalla porta principale, ideale per chi desidera approfondirne la conoscenza e acquisire in tal modo una professionalità sempre maggiore.